# MOTION DETECTION FOR PC BASED ON SECURITY SYSTEM BY USING OPTICAL FLOW

**NUR NABILAH BT MOHAMAD HAMID**

**BACHELOR OF ELECTRICAL AND ELECTRONICS ENGINEERING**
**UNIVERSITI MALAYSIA PAHANG**

## UNIVERSITI MALAYSIA PAHANG

# BORANG PENGESAHAN STATUS TESIS♦

**JUDUL**: **MOTION DETECTION FOR PC BASED SECURITY SYSTEM BY USING OPTICAL FLOW**

**SESI PENGAJIAN**: **2010/2011**

Saya **NUR NABILAH BINTI MOHAMAD HAMID ( 880906-04-5204 )**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan ( √ )

| | | |
|---|---|---|
| [ ] | **SULIT** | (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) |
| [ ] | **TERHAD** | (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan) |
| √ | **TIDAK TERHAD** | |

Disahkan oleh:

| | |
|---|---|
| (TANDATANGAN PENULIS) | (TANDATANGAN PENYELIA) |
| Alamat Tetap: | **DR. KAMARUL HAWARI BIN GHAZALI** |

**KM 1018, JLN MUTIARA 2,**
**TAMAN MUTIARA,**
**78000 ALOR GAJAH,**
**MELAKA**

Tarikh: **26 NOVEMBER 2010** Tarikh: : **26 NOVEMBER 2010**

CATATAN: * Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : ______________________________

Name : DR. KAMARUL HAWARI BIN GHAZALI

Date : 26 NOVEMBER 2010

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ___________________________________

Author : NUR NABILAH BT MOHAMAD HAMID

Date : 26 NOVEMBER 2010

*To my beloved father mother and the greatest parent ever,*
*Mr Mohamad Hamid bin Nasir and Rosidah bt Sidek.*

*To my supportive and awesome sibling,*
*Nur Nadirah bt Mohamad Hamid;*
*Nur Nasiha bt Mohamad Hamid,*
*Muhammad Naqqiuddin bin Mohamad Hamid*
*and*
*Muhammad Ariff Nadzaruddin bin Mohamad Hamid*

# ACKNOWLEDGEMENT

First of all, I would like to thank Dr Kamarul Hawari bin Ghazali, who introduced me to the fascinating world of Image Analysis and Computer Vision and for his help and supervision during the development of this thesis. All of his advice, idea and the time he spent discussing for some of the problems I had while working on this project and thesis.

I further want to thank all of my friends for their support and the wonderful time we had as well as spend their while helping me solve problem occur during done this project.

And last but not least, I thank my parents for their ultimate support during all of these years.

# ABSTRACT

This system will detect and analyze the motion of people that recorded on PC then gives the feedback on the spot if there have an abnormal motion. Horn-Schunck method is one of optical flow method which is a method or technique to detect the motion in an image sequent. By this method, the we can know the velocities of motion object in the image sequent. Afterward, velocities will be analyze to determine object movement. Hence, if there is abnormal motion, this system will give the alert. The efficiency of this sytem are 81%.

# ABSTRAK

Projek ini membina satu sistem bagi pengesan pergerakan melalui komputar persendirian berdasarkan sistem keselamatan dengan menggunakan Aliran Optik. Sistem ini akan mengesan dan menganalisis pergerakan manusia yang dirakam dari komputer persedirian. Kaedah Horn-Schunck merupakan salah satu kaedah Aliran Optik di mana ianya merupakan kaedah atau teknik untuk mengesan pergerakan di dalam satu urutan gambar. Melalui kaedah ini, kita akan mengetahui kelajuan pergerakan objek yang terdapat di dalam urutan gambar tadi. Kemudian, nilai kelajuan akan dianalisis untuk menentukan pergerakan objek. Maka, jika terdapat pergerakan tidak normal, sistem ini akan memberikan amaran. Kecekapan sistem ini adalah sebanyak 81%.

## TABLE OF CONTENT

# CHAPTER 1

# INTRODUCTION

## 1.1 Project Background

This project concerns with the development of a system whereby motion detection for PC based security system using optical flow. One of the product or method for security system is closed-circuit television (CCTV). The existing CCTV is only for monitoring and observe as well as record the activities. My project is a system that put some intelligent on the function of closed-circuit television (CCTV).

This system will detect normal and abnormal motion of the a movement that recording by CCTV from PC by use algorithm technique development using optical flow. Then, this system will give the feedback on the spot by giving alert. Alert will be given if there is any detection of abnormal motion. Normal motion is consider as people walk. In contrast, abnormal motion consider people run at the same place or place.

## 1.2 Optical Flow Method

Optical flow is a useful method in the object tracking branch and it can calculate the motion of each pixel between two frames, and thus it provides a possible way to get velocity of the motion object [1]. It is the apparent motion of the brightness pattern in an image sequence.

Optical flow technique is the most popular technique using by engineer to detect motion. This project will use one of optical flow method which is Horn-Schunck method. This method is global approach that more sensitive to noise. By using this method, we can evaluate the velocity of the pixel moving across the one image into the next image in a sequence.

## 1.3 MATLAB

This project use MATLAB software to apply the optical flow method. This software is high-level technical computing language and interactive environment for algorithm development, data analysis, and numeric computation. Applying optical flow method program, the image capture by CCTV can be analyzed by using this software. So, we can know either there has abnormal or normal motion from the a movement that capture by CCTV from the analysis.

## 1.4 Problem Statement

The existing CCTV is only for monitoring and observe as well as record the activities, it does not give a lot of help while any crime happen. Criminals still free outside there because there are a lots of crime never get the solution. Usually, there will be delay in action when the crime occur such as police will arrive after the crime happen.

Then, to find the criminal, the scene that record by CCTV will be analyze and diagnosis. While waiting for result of the diagnosis, there are a lot of procedure waiting for result approval and without knowing, criminal will done more crime or they can grab the opportunity to find a solution that is not charged as a criminal.

In order to analyze the recorded activities as well can give the result on the spot, optical flow method use to analyze the motion. This method can be use to differentiate the movement pattern. The pattern of criminal movement are different from normal person movement. So, the movement pattern can be classify into two categorize which are abnormal and normal motion. An analysis should be done to differentiate the movement pattern.

## 1.5 Objectives

The objectives of the project that need to be achieved are:

i. To detect the suspicious and abnormal motion from a movement using optical flow method
ii. To apply optical flow in image processing analysis

## 1.6 Scope of Project

The scopes of this project are:

i. To apply the optical flow method which is Horn-Schunck method in MATLAB software.
ii. To relate the calculation using optical flow with the motion image.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

This chapter will covered and summarize the content of the paper or presentation of conference, journal, research, report and any article that were studied related to the project. The content will give some overview about system and method that will use in this project. Some of studies give references in order to develop this project.

## 2.2 MATLAB

MATLAB stand for matrix laboratory is a technical computing environment for high-performance numeric computation and visualization. This software integrates numerical analysis, matrix computation, signal processing, and graphics in an easy-to-use environment where problems and solutions are expressed just as they are written mathematically which is without traditional programming [2].

This software also provide a product that acquire, process, and analyze images and video for algorithm development and system design for image and video processing.

### 2.2.1 Pre-processing

The pre-processing performs some steps to improve the image quality or to enhance the data. This steps is needed to reduce the project assumption. For instance, if the project need to analyze iris but in the image not only the iris, but also some useless parts such as eyelid and pupil. Then, the camera to eye distance may also cause variations the size or angle or position of the same iris. Beside, image captured in low resolution, so it is hard to analyze the image as well as affect feature extracting. Therefore, it is necessary to do pre-processing the original images.[3]

### 2.2.2 Data acquisition

The first stage of any vision system is the data acquisition stage. The data can be image and video. After the image has been obtained, various methods of processing can be applied to the image to perform the many different vision tasks required. However, if the image has not been acquired satisfactorily then the intended tasks may not be achievable, even with the aid of some form of image enhancement [4]. Images are typically generated by illuminating a scene and absorbing the energy reflected by the objects in that scene.

### 2.2.3 Image processing

In computer vision, image processing is any form of signal processing for which the input is an image, such as a image or video (extract into frame). The output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two dimensional signal and applying standard signal-processing techniques to it.

In many cases, image processing concerned with taking one array of pixels as input and producing another array of pixels as output. This same as improving the array from the original array. By processing the input or data, this process may remove noise, improve the contrast of the image, removing blurring that caused by movement of the camera during image acquisition and as well as correct for geometrical distortions caused by lens [4].

### 2.2.4 Feature extraction

Feature extraction is a very important field with growing applications in science and engineering. The main aim of feature extraction is to extract important features from image data, from which a description, interpretation or understanding of the scene can be provided by the machine. When the input data to an algorithm is too large to be processed, then the input data will be transformed into a reduced representation set of features. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy [5].

### 2.2.5 Classification

Image classification can be done after extract some features out of the image. Its analyze the numerical properties of various image features and organizes data into categories. In order to classify the image, two process are needed which are training and testing. Initially, training phase, characteristic properties of typical image features are isolated and, based on these, a unique description of each classification category is created. In the subsequent testing phase, these feature-space partitions are used to classify image features.[6] To demonstrate the performance of the learning algorithm, random data drawn from different classes were generated and used the proposed method to learn the parameters and to classify the data. [7]

## 2.3 Security system

Security system needed to ensure a safe places to live and used either in public or private. Airport, shopping center, financial institution, hospital and education institution are the area has a lot of people and crime occur easily. So, an efficient and good security system level play important role. In 2008, conference about Safety and Security System in Europe demonstrate exist security system that upgraded using Artificial Intelligent system.

Security guard responsible to protect or prevent crime happen, but their have limitation due to human strength and not able to describe criminal look like detailed [8]. Beside, not all security guard has certificate or knowledge or training about security. Closed-circuit television was developed initially as increasing the security and can give more  information about crime happened.

### 2.3.1 Closed circuit television

Close circuit television (CCTV) security cameras play important role to give fairly strong and consistent evidence [9]. The images that we see from our eyes can as good as images that record by CCTV security cameras. The specification factor can effect the quality of camera images. Alarms can be embedded with CCTV in order to make the security system more efficient. The contribution made by combination of alarms and CCTV may be potential to increase numbers of detected cases [10].

The applications of image processing in CCTV make this system become more intelligent. Beside, use of image processing enables enhancements to many aspect such as attempt to avoid incident, detect ongoing incident in time to enable intervention as well as collect the evidence for post incident use. Also, improve edge enhancement, for instance, make highlighted objects more easy recognizable. Make more stable in auto tracking white balance and not easy influenced by color objects in the scene [11].

## 2.4 Optical flow

The detection of moving objects is critical in many defense and security applications, where motion detection is usually performed in a pre-processing step, a key to the success in the following target tracking and recognition. Many videos used in defense and security applications are out-door videos whose quality may be degraded by various noisy sources, such as raining and wind [12].

Motion detection has been extensively investigated. In computer vision, motion is an important queue in order to used in tracking, structure from motion and video compression. Motion cannot be observe directly but we can observe image and see how points in the image move [13].

As one of the major techniques, optical flow-based approaches have been widely used for motion detection [12]. Optical flow will compute the velocity of the motion object between two consecutive frames of an image sequence. An image sequence is an ordered set of images and the velocity of the motion introduced in such an image sequence [14].

Optical flow is a method to detect motion which is a calculation method to detect the motion based on brightness and spatial smoothness. Optical flow-based approaches have been widely used and popular method use by engineer for motion detection [12]. So, it has a long history and assumptions of brightness constancy and spatial smoothness underlie most optical flow estimation methods [15].

Optical flow can arise from relative motion of objects and viewer, so it could give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement [16]. Basically, optical flow is a velocity field of the image generated from the transformation of one image into the next image in a sequence [17]. Motion perceived when a changing picture is projected onto a stationary screen [17].

### 2.4.1 Optical flow exploration

Optical flow is a useful method in the object tracking branch and it can calculate the motion of each pixel between two frames [1]. A long history of optical flow estimation make have a lot of optical flow method have explored some variation of the same theme [15]. There are a lot of exploration describing the implementation of optical flow. Some result are acceptable, but in many project, there are limitations.

Most of the method or techniques exploit the two same constraints which are brightness constancy and spatial smoothness. The brightness constancy constraint is derives from the observation that surfaces usually persist over time and hence intensity value of a small region remains the same despite its position change. The spatial smoothness constraint come from the observation that neighboring pixel generally belong to the same surface and so have nearly the same image motion [15].

Since optical flow has been introduce, there have been a few effort to learn about the brightness constancy and spatial smoothness [15][18]. Recently, an adequately realistic image sequences with ground truth optical flow have been made and finally make this practical. A number of classic and recent optical flow has been revisit and the training data and machine learning methods that can be used to train has shown. From the advances have been made, there is a research go beyond previous formulations to define new versions of the data and spatial terms [15].

Recently advances have made two primary contribution which are exploit image intensity boundaries to improve the accuracy of optical flow near motion boundaries and learn a statistical model of the data term. The idea to exploit image intensity boundaries is based on Nagel and Enkelmann who introduced oriented smoothness to prevent blurring of flow boundaries across image boundaries. The common brightness constancy assumption have addressed as a problems by several authors make the research want to learn about statistical model.[15]

The local image edge orientation was use in order to define a steered coordinate system for the flow derivatives. The flow derivatives along and across image boundaries that highly kurtotic are highlighted. Normally, the spatial smoothness of optical flow is expressed in term of the image axis aligned partial derivatives of the flow field. By using Markov random field and Gaussian scale mixture in flow field, a rigorous statistical formulation of the idea of Nagel and Enkelmann.[15]

With the intention of minimize the effects of illumination change, brightness constancy were extended to high-order constancy, such as gradient and Hessian constancy [2][19]. Furthermore, a research show that improving the accuracy of dense optical flow by integrating constraints within a local neighborhood [15][20]. Additionally, Field-of-Expert formulation has extended to the spatial-temporal domain to model-temporal changes in image features [15][18].

### 2.4.2 Problem and issues in optical flow

There are much progress has been made in optical flow computation and some issues have been made due to numerous theoretical and practical reasons [21]. Theoretically, optical flow is an approximation to image motion and largely determines the lower bound on accuracy. Beside, the scene properties such as occluding surface also make an issue in optical flow computation. Practically, based on a study by Barron et al in 1994, of computing optical flow that analyzed nine techniques based on accuracy and reliability of measurements[21][22].

Optical flow often misunderstood between image motion. The optical flow differ from image motion based on some conditions. In the case of the absence of texture, optical flow is zero and when true motion field violates the brightness consistency model used for its approximation [17][21]. Current optical flow methods mostly assume the uniform scene illumination and Lambertian surface reflectance are either explicitly or implicitly which use some form of the brightness consistency assumption [21].

Occlusion is difficult to analyze, despite the fact that occlusion constitutes an important source of visual information. The fact that image surfaces may appear or disappear in time, misleading tracking process and causing numerical artifact in intensity derivatives make occlusion difficult to handling. These problems have been express by research community. In optical flow, to determine the direction of translation and segment the scene into independently moving surface, the occlusion boundaries may be used [21]. Most of optical flow techniques relied on a single surface hypothesis which a rare visual event [17][21].

Hierarchical correlation methods is robust motion measurement schemes for image sequences with significant contrast change or large displacement and severe aliasing. Based on a study by Barron et al in 1994, the test image sequences used are all appropriately sampled with small motions (between one and four pixels per frame) and favorable to differential approaches. From their study, correlation methods experience difficulty with sub-pixel motions as their error depends on the closeness of the image motion to an integer number of the pixels. So, as the alternative of correlation methods is Hierarchical differential-based methods [22].

In case to threshold optical flow field or to weight velocities in post measurement processing such as in a motion and structure calculation, the confidence measures can be used. Confidence measures are needed to indicate the reliability of computed velocities [21]. Based on a study by Barron et al in 1994, the smallest eigenvalue of a least-square matrix were use although most of differential methods do not provide confidence measures [21][22].

### 2.4.3 Optical Flow previous work

Optical flow method have a lot of technique that introduced by researcher. Black and Anandan introduced a robust estimation framework to deal with such outliers, yet did not attempt to model the true statistics from example. Fermiller et al analyzed the effect of noise on the estimation of flow. However, they did not attempt to learn flow statistics of brightness computation [15].

Horn and Schunck introduced both the brightness constancy and the spatial smoothness constraints for optical flow estimation. In spite of this, their quadratic formulation assumes Gaussian statistics and is not robust to outlier caused by reflection, occlusion and motion boundaries as example [15][17].

The brightness constancy assumption have extended by many authors, either by making it more physically plausible or by linear or non-linear per-filtering of the images [15]. Recently the idea of assuming constancy of first or second image derivatives to provide some invariance to lighting changes with Laplacian pyramid has been updated [19]. The idea is connected by replaced the pixel-wise brightness constancy model with spatial smoothed one [20].

There are some problem encounter at motion boundaries where the assumption of spatial smoothness is violated. Nagel and Enkelmann introduced oriented smoothness to prevent blurring of optical flow across image boundaries by observing that flow boundaries often coincide with image boundaries. Then, some modification have made so that, less smoothing is performed close to image boundaries. Beside, the amount of smoothing along and across boundaries has been determined heuristically [15].

Lately, the spatial structure of optical flow field using a high-order MRF, called a Field of Expert and learned the parameters from training data. Roth and Black combined their learned prior model with a standard data term and found that

Field of Expert model improved the accuracy of optical flow estimation [15][18][20]. They only models the spatial statistics of the optical flow and not the data term or the relationship between flow and image brightness although they provide a learned prior model of optical flow [15].

## 2.5 Optical flow techniques

A survey had done for the optical flow classes. There are six optical flow classes which are intensity-based differential methods, multi-constraint methods, frequency-based method, correlation-based methods, multiple motion methods and temporal refinement methods. The boundaries between each class of methods are not always clear [21].

Weng's method incorporates both phased-based and feature-based matching while Waxman et al's applies a differential scheme on time varying edge maps. However, both method were classify as a phase-based method and the latter as a differential method. Furthermore, multiple motion and temporal refinement methods were overlap with other classes. But, their importance dictates that they be covered separately [21].

### 2.5.1 The optical flow constraint equation

The optical flow cannot be computed at a point in the image independently of neighborhood points without introducing additional constraints, because the velocity field at each image point has two components while the change in image brightness at a point in the image plane due to motion yields only one constraint.

Assumption made by most of the optical flow method is that the intensity $I$ of moving points is constant over a period of time. This is known as brightness constancy assumption.

$$I(x, y, t) = I(x+dx, y+dy, t+dt) \tag{2.1}$$

Thus, the optical flow constraint equation (OFCE) is obtained by using Taylor expansion in (2.1) and dropping its nonlinear terms. Therefore, the OFCE can be expressed in the form as

$$I_x u + I_y v + I_t = 0 \tag{2.2}$$

Where *(u, v)* represent the optical flow vectors (dx/dt, dy/dt) and $(I_x, I_y, I_t)$ represent the derivatives of image intensities at coordinate *(x, y, t)*

## 2.6 Differential methods

Image velocity form spatio temporal derivatives of image intensities are compute by differential technique. The image domain is therefore assumed to be continuous or differentiable in space and time. Global and local first order and second order method can be used to compute optical flow [21].

Global methods use an additional global constraint which is usually a smoothness regularization term in order to compute dense optical flows over large image region. While, local methods use normal velocity information in local neighborhoods to perform a least squares minimization to find the best fit for velocity, ***v***. The size of neighborhood for obtaining a velocity estimate determines whether each individual technique is local or global [21].

## 2.7 Optical flow algorithms

There are two classic methods of optical flow computation in computer vision which are Horn-Schunck method and Lucas-Kanade method. Both of them were invented in 1980's and they are easily to be implemented [1]. Optical flow also consist two catogaries which are local method and global method. Lucas-Kanade is as example of local method while Horn-Schucnk and Brox et al are global method.

Unfortunately, Lucas-Kanade method may not perform well in dense flow field, on the other hand, Horn-Schunck method can detect minor motion of objects and provide a 100% flow field [12][20].

### 2.7.1 Lucas-Kanade method

The Lucas-Kanade was introduced by Bruce D.Lucas and Takeo Kanade. In computer vision, this method is a two-frame differential method for optical flow estimation. It introduces an additional constraint that the optical flow is varying smoothly long the neighboring object points that possessed the same velocity.

The Lucas and Kanade algorithm is a solution of image registration. Image registration has a variety of applications in computer vision, such as, image matching for stereo vision, pattern recognition, and motion analysis. Existing techniques for image registration tend to be costly, and they fail to deal with image rotation and distortions. The Lucas and Kanade algorithm presented a new method that uses spatial intensity gradient information to direct the search for the position that yields the best match. This is how the Lucas and Kanade algorithm came about [1].

### 2.7.2 Horn-Schunck method

This method introduced by B.K.P Horn and B.G. Schunck. This method will use Laplacian operation. Horn and Schunck define optical flow is a velocity field in the image that transforms one image into the next image in a sequence [16]. This method is based on the two-frame differential algorithms.

It is a special approach of using global constraint of smoothness to express a brightness variation in certain areas of the frames in a video sequence [12]. To avoid variations in brightness due to shading effects, objects are assumed to have flat surfaces. The illumination across the surface is assumed to be uniform. Horn and Schunck assumed that reflectance varies smoothly and has no spatial discontinuities. This assures them that the image brightness is differentiable [1].

### 2.7.3 Brox et al. method

This method is a famous method and more sophisticated method. It is a novel approach that integrates several successful concepts and addresses all the problems of the Horn-Schunck model [14]. Its extend brightness constancy to high-order constancy, such as gradient and Hessian constancy in order to minimize the effects of illumination change [15].

### 2.7.4 Proesmans et al. method

Proesmans et al. present a method similar to that of Horn and Schunck. The main differences are that a matching process is incorporated into the constraint equation and a method for dealing with discontinuous flow is introduced.

To evaluate the current flow estimate, they look at the correspondences suggested by the current flow estimate. For each point, $(x, y)$, in the first image the estimated optical flow, $(u, v)$, can be used to find a corresponding point,$(x+u, y+v)$, in the second image. If the flow estimate is good then the brightness at these two points should be similar. If the brightnesses are not the same then the flow estimate is moved along the image gradients in order to correct for the difference.

A process known as anisotropic diffusion is used to smooth the flow. This means that the local averages of the flow components are not simple averages but are weighted by a consistency map. Unlike the normal smoothing process, this can maintain discontinuities in the flow but still retains a high level of continuity in local regions [23].

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

This chapter will review the flow or methodology of this project from the very start until this project complete.

## 3.2 Methodology

This project involved in development of the program by using MATLAB software and the data source from video recording that similar as CCTV recording. To achieve the objectives of this project, methodology were construct base on the scopes of project. A terminology of works and plans show in the flow chart. This is very important to make sure the goal and purpose of this project can be achieved.

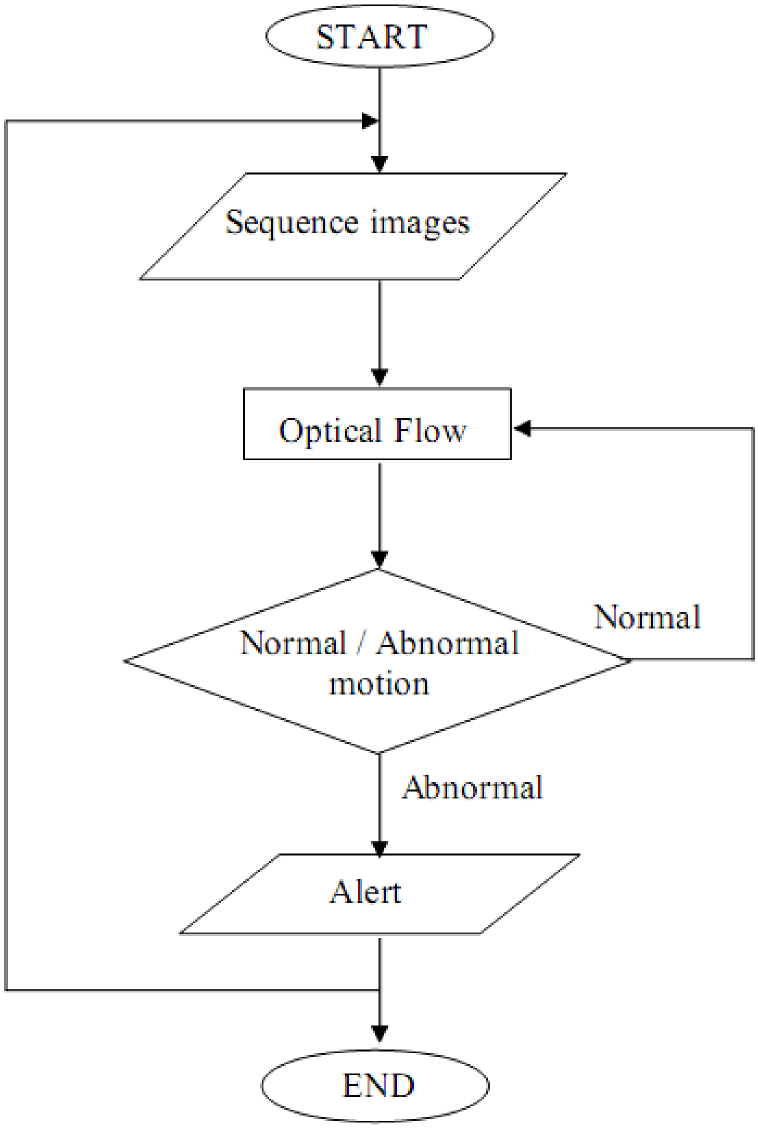



**Figure 3.1:** Flow for overall project


**Figure 3.2:** Flow while doing Image Processing analysis

## 3.3 Initial assumption and data analysis

Before started this project, the question that appear is how to analyze person based on their motion or movement?. Base on this project, Optical Flow method was use had decide Horn-Schunck method was choose. So, Horn-Schunck method that used in this project will calculate the velocity each pixel for two different image in sequence. Horn-Schunck method was constructing as a function in MATLAB.

The output from Horn-Schunck method produces two variables which are *u* and *v*. Where *u* is flow at *x-axis* and *v* is flow at *y-axis* and both of this variable will be represent as arrow. So, the value of velocity can be calculated from the value of *u* and *v* using equation below:

$$velocity = \sqrt{u^2 + v^2} \tag{3.1}$$

Then, another question appears which is how difference the velocity of person activities? Walking person should have low value of velocity than running person. In order to analyze the image, statistical method were use. The mean, variance and standard deviation of velocity values for two difference image where produce to evaluate differentiate its value for difference pattern of movement.

This project started by analyze image with assuming person moving with a ball moving image. The moving ball images were created by using paint software that provide by window. There are some precautions to be aware to ensure initial outputs of analysis are strong enough as reference for the real image analysis.

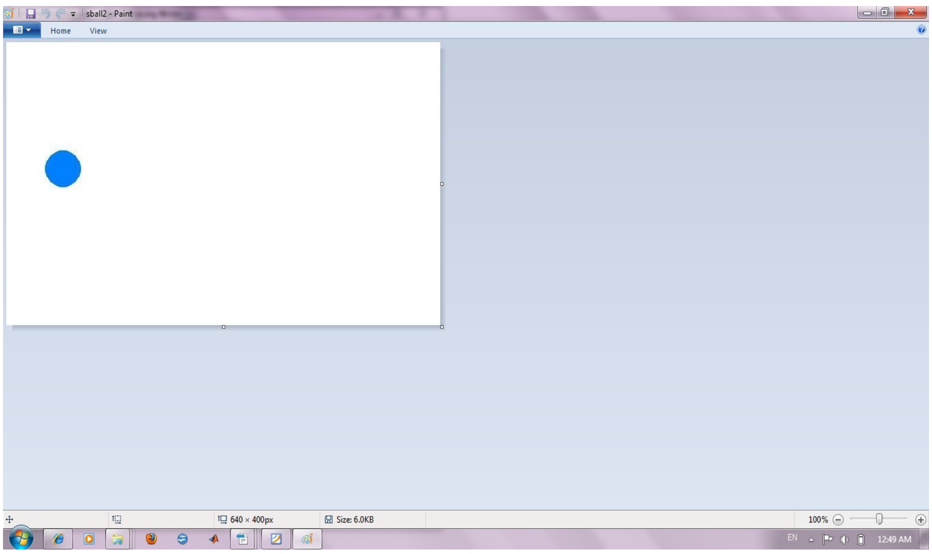



**Figure 3.3:** Creating data using paint software

The precaution are ensure the image produce are in 640x420 pixels, the size of ball are same for each different image, all image were save in same format which is JPEG format. From the image created, there are two conditions were made. First, ball movement in small distance considers as people walking and long distance can be considered as people running.

It is because by considering the distance and time, people walking will produce sequence images where the object position in image two closes with object position in image one (small distance). On the other hand, people running will produce sequence images where the object position in image two far from the object position in image one (long distance).

Then, the maximum, minimum, mean, variance and standard deviation of velocity value were analyzed. From the observation, minimum and maximum cannot be use as parameter to analyze and it is not suitable as well cannot give appropriate value to differentiate for small and long distance.

The mean, variance and standard deviation give value that are use to analyze. Those parameters give the difference value for both small and long distance and will be considered as a parameter for this project. In this project just use only one parameter to analyze the motion.

After do some studies in mathematics subtopic which are probability and statistics, the meaning and application of mean, variance and standard deviation were can be clearly understood. Mean is the average of the numbers which add up all the numbers, then divide by how many numbers there are. The Standard Deviation ($\sigma$) is a measure of how spread out numbers is. The Variance ($\sigma^2$) is defined as the average of the squared differences from the Mean.

The output of Horn-Schunck method is the value of velocity for each movement that happen for each pixels. Then, to get the velocity for whole image, the mean value are useful. Compare to variance, standard deviation value is more useful. Standard deviation very useful while doing classification. For example in order to evaluate people height, standard deviation will tell us who the taller and shorter one based one mean value. Hence, to analyze the velocity for whole image, mean value where use.

Before analyze the image, there are some assumption were made. For walking motion, the value of mean of velocity must be lower than running motion. For random motion, the value of mean of velocity must be not constant which its value must be drastically low or high. The last assumption is for the static motion, the value of mean of velocity will be zero (0).

These assumptions were prove false in the analysis. It is because small distance moving object, the mean value are higher than long distance moving object. From the analysis, an idea can make which person's walking will produce high value on mean of velocity while person's running will produce smaller value on mean of velocity.

An idea come out based on the observation that, these happen because of in small distance, because of the object move very slow, the motion can be detect clearly based on the are of object shown and the Horn-Schunck method produce more arrow for it output. On the other hand, from long distance, the object moves too faster to detect the motion.

Beside, another condition was made which is the ball moving randomly. This condition will consider as people move randomly and this condition can be consider as abnormal motion. The result show the value of velocity's mean also not consistence which is the value can be high and so suddenly it can be low. For static motion, there is zero value of mean of velocity produce. This random movement will be analyzing as further additional analysis if needed. For this project, walking and running will be main analysis.

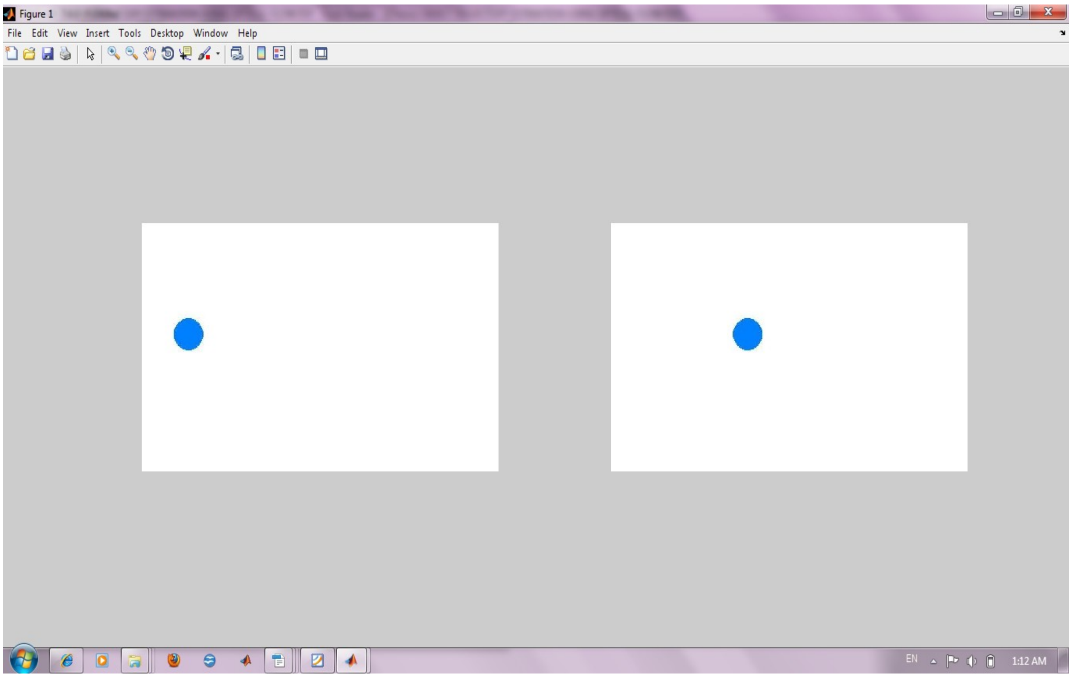



**Figure 3.4:** Two difference image with small distance moving object

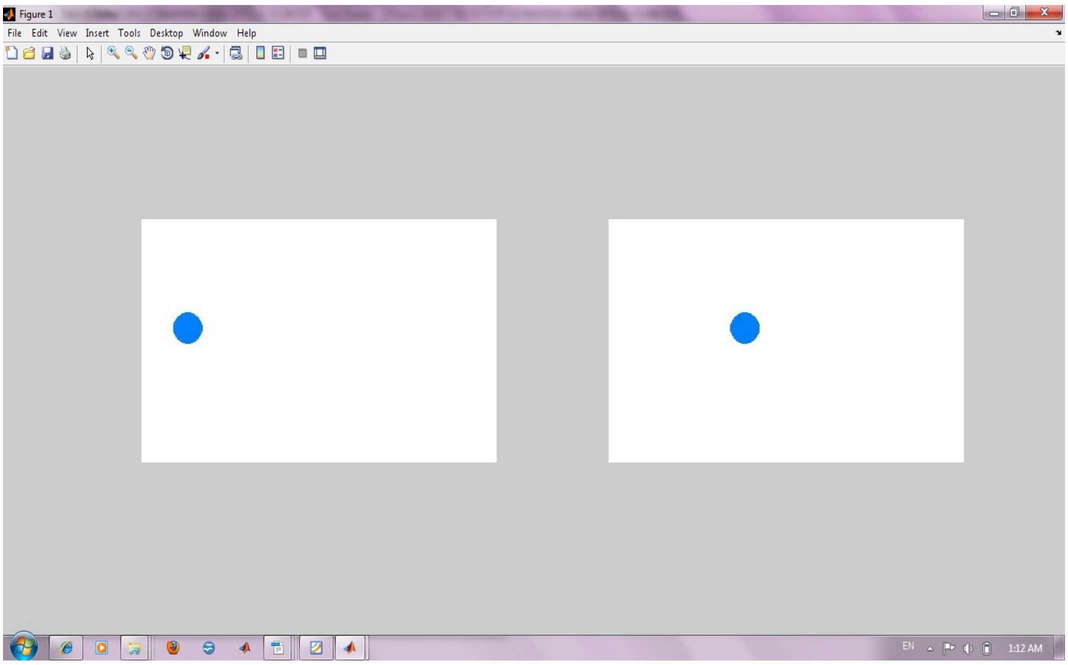



**Figure 3.5:** Two difference image with long distance moving object

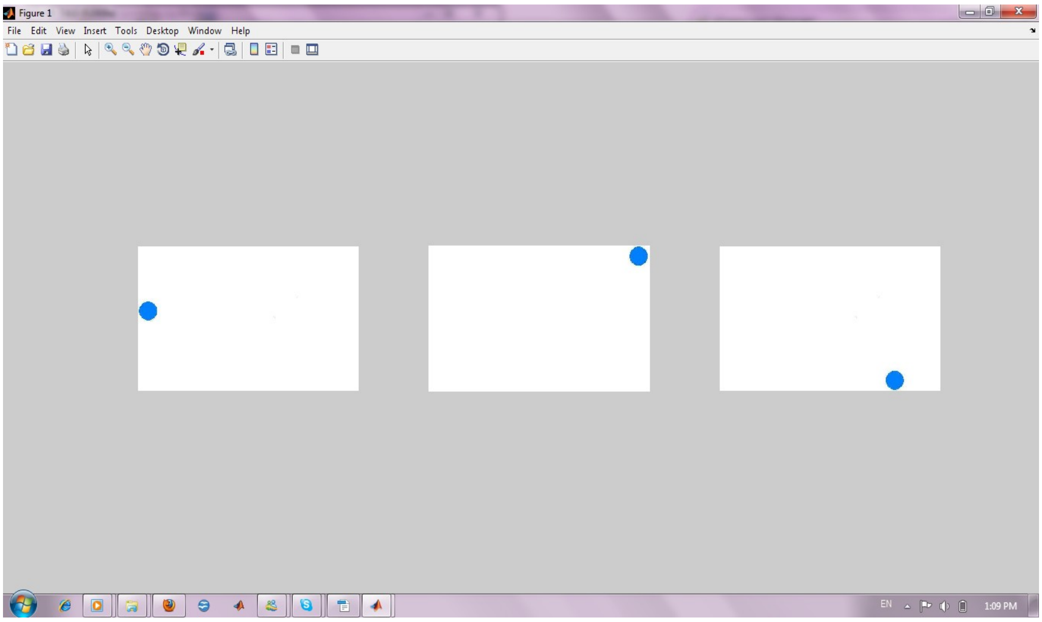



**Figure 3.6:** Image sequence for randomly moving object

Furthermore, in order to making this system more efficient, looping technique was use to differentiate multiple image in sequence. This looping technique was use to differentiate two image which the reference image was fix. So, the first image will be compare with it next image and so on. To collect the mean value for each loop, the function of fopen, fprint and fscan were use and also to produce the graph mean versus image number.

The function of fopen was use to open the notepad path to where it will use to put the list of mean value. Furthermore, to put the mean value, fprint function will use and lastly to produce the graph, the data will be read from the file. So, fscan function is needed.

After done looping technique and graph were produce. The value of velocity's mean can evaluate through graph for a sequence of images. The graph show the mean value versus image number sequence. From the graph shown, there have differences in term of graph pattern.

## 3.4 Real analysis

After get the initial output value as reference, this project proceed by using the real situation of object movement. To develop the real situation image, this project will use video that capture people walking and run. Then this data will be analyze and to compare with the initial output. From this procedure, it will make the analysis output be stronger and the next level of image processing step can be made which is classification. The image sequent will contain from 10 to 42 images in a sequence.

### 3.4.1 Pre-processing and data acquisition

In order to analyze the real situation, a video that capture by CCTV of the moving people will be analyze using optical flow algorithm. In this project, based on optical flow method which evaluate the velocity of the pixel moving across the one image into the next image in a sequence, the video as project data collection, will be convert to image from the very start before use the data.

Video to JPG converter software was use as the converter. While use this converter, there is converter performance need to be setting. For this project, in order to make the data collection will be constant or as controlled variable, the video were extracting into 5 frames per second.

### 3.4.2 Assumption making and feature extraction

This project develop by analyze the movement of person. The movement was divided into two categories which are walk and run. Then, the movement will be classified either in abnormal or normal motion. Table below show the classification of the movement:

| Normal Motion | Abnormal Motion |
|---|---|
| - Walk | - Run |

**Table 3.1:** Table for movement classification

This movement will be analyzing based on their velocity value. The value of means of velocity in two difference image can be evaluated. The value of velocity is linear with the value of mean of velocity. Higher velocity, the mean value will be high. However, the motion cannot be categorize without making some additional consideration.

### 3.4.3 Increase system efficiency and classification

To make this project more efficient, the moving object from image background should be eliminate to prevent the velocity from the background effect the value of velocity of real moving object. Sixteen sub-block of the velocity was done. So that, mean of velocity for each sub-block will be evaluate.

**Figure 3.7:** Block of velocity before become sub-block

| | | | |
|---|---|---|---|
| A11 | A12 | A13 | A14 |
| A21 | A22 | A23 | A24 |
| A31 | A32 | A33 | A34 |
| A41 | A42 | A43 | A44 |

**Figure 3.8:** Sub-block of velocity

But then, problem occurs because it is difficult to evaluate mean value for each sub-block. Hence, to solve it, division of velocity in programming was done and the velocity for each sub-block can be evaluated by calling the row and column to evaluate. From the mean of velocity value list form each sub-block were observe and a threshold can be made in term of mean value. This threshold will determine which row and column have main motion or background motion.

A classification can be make by considering the mean value for any row and column that below from 1, will be change into 0 or in other word, that part will be eliminate from evaluation for motion image. It is because by consider the mean value below than 1 are mean value for background motion. But then, after do more detail analysis in term of velocity value for each row and column, the new threshold value of mean that to be eliminating change to 2.

Finally, because of the threshold value for mean value for each row and column had change into 2, if the mean value less than 2 will be consider into 0 or in the other word, eliminated. On the other hand, if the mean value greater than 2 will be not change and its will be the mean value of main motion. After this threshold technique, the new mean value which is without mean value from background motion will be use as further analysis.

Then, the value of exist mean which is new value of mean can be evaluate to classify either it is normal or abnormal motion. After doing summation for every mean value for each row and column, the new mean value will be produce and classification can be made. In order to decide the threshold value, the value of mean and the graph pattern has been observe.

Before decide the threshold value, some image sequence need to be train to get the proper threshold value. There are 2 difference of image sequence input which are running image sequence and walking image sequence. The result show that, mean value for running image sequence are mostly around 18 to 25 while for walking image sequence, the mean value mostly around 25 to 42. So that, the threshold value to classify and differentiate both motion decided to be 25.

Furthermore, if the mean value below than 25, it consider as run. While, greater than 25, it will be consider as walk. So far, from the initial analysis result can be considered as prove base on the result on the real analysis result.

### 3.4.4 Decision making

A problem occurs when the image sequence of running also detected as walking and vice verse. Hence, to ensure the image sequence is walking or running image sequence, a image sequent indicator were created. This indicator will detect how many sequent of image while do classification before make decision. If the system indicates 5 in sequent of running or walking in one images sequence, the system will tell either the sequence is running or walking images sequence. Furthermore, decision can be made either there have abnormal motion or not. If there has abnormal motion, the system will be signal of alert on command window.

Later, to ensure there are no internal or error while indicate the 5 sequent images, the program that indicate numbers of image sequent will be reset for both running and walking image sequent in this system programming if the image detect different image. For example, this system indicates there is 2 walking image and so suddenly this system indicate running image. So, the value that indicate 2 for walking images will be reset and become 0 and same goes to running image sequent.

After done all of the analysis, assumption and adding some conditions on the system programming, this system can decide the movement pattern based on mean of velocity value through the people motion in a images sequence. Setting up the threshold value on the system programming can make this system to give the appropriate output of this system.

This system will indicate the image sequent as well mean value and give the decision. In this system if the value is higher from the threshold value, this system will produce an output and tell the image sequence is running but then if lower than threshold value, the image sequence is walking.

### 3.4.4 System efficiency

This project proceed by ensuring how much efficient this system, this system training by using difference running and walking image sequence. This image sequence will be difference walking and running of difference person. Each person will running and walking. Their image sequence will be observed. Thus, the efficient of this system can be determined.

The efficiency of this system will be evaluated with 5 sequences of both, walking and running images. The variable that will be evaluated is the last condition which is to detect the number of walking or running image before decide either in the sequence is walking or running image sequence.

In order to get the efficiency, every image sequence was train in this system programming and the output will be observe to get efficient value for this system. Every image in an image sequence will be training and count how many times this system give the wrong decision. The image sequent indicator will be change in order to get which indicator is the best and efficient for this system. The indicator change show in table 3.2.

| | **run** | **walk** |
|---|---|---|
| **Indicator condition** | 5 | 5 |
| | 5 | 3 |
| | 3 | 5 |
| | 3 | 3 |

**Table 3.2:** Image sequent indicator setting

Beside, by testing the image sequence and find out the efficiency, 4 image sequences of running were only 10 images from each sequence will be test. But then, 4 image sequences of running were 30 images from each sequence will be test.

# CHAPTER 4

# TECHNIQUE

## 4.1 Introduction

This chapter will discuss about the techniques use in this project. All of these techniques used to make this programming for this system more efficient and reduce memory space.

## 4.2 Horn-Schunck technique

There are two classic methods of optical flow computation in computer vision: Horn-Schunck method and Lucas-Kanade method [3][4][5]. Both of them are based on the two-frame differential algorithms. Lucas-Kanade method may not perform well in dense flow field. On the other hand, Horn-Schunck method can detect minor motion of objects and provide a 100% flow field [3]. Thus, this project focus on Horn-Schunck method for optical flow computation.

As mentioned earlier, to detect and analyze motion in two difference image in sequences, one of the optical flow method will be use which is Horn-Schunck method. In order to generate the output from optical flow, Horn-Schunck method function will be call as shown below and follow by the function of Horn-Schunck method description. This function will be call earlier for propose to analyze the function output.

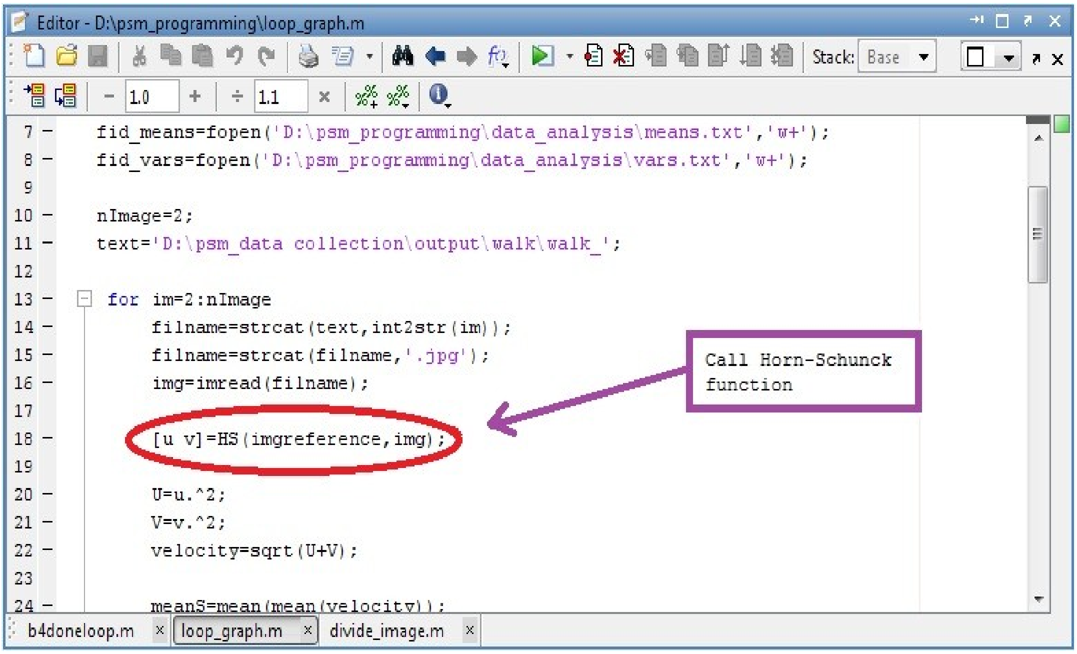



**Figure 4.1:** Call Horn-Schunck function

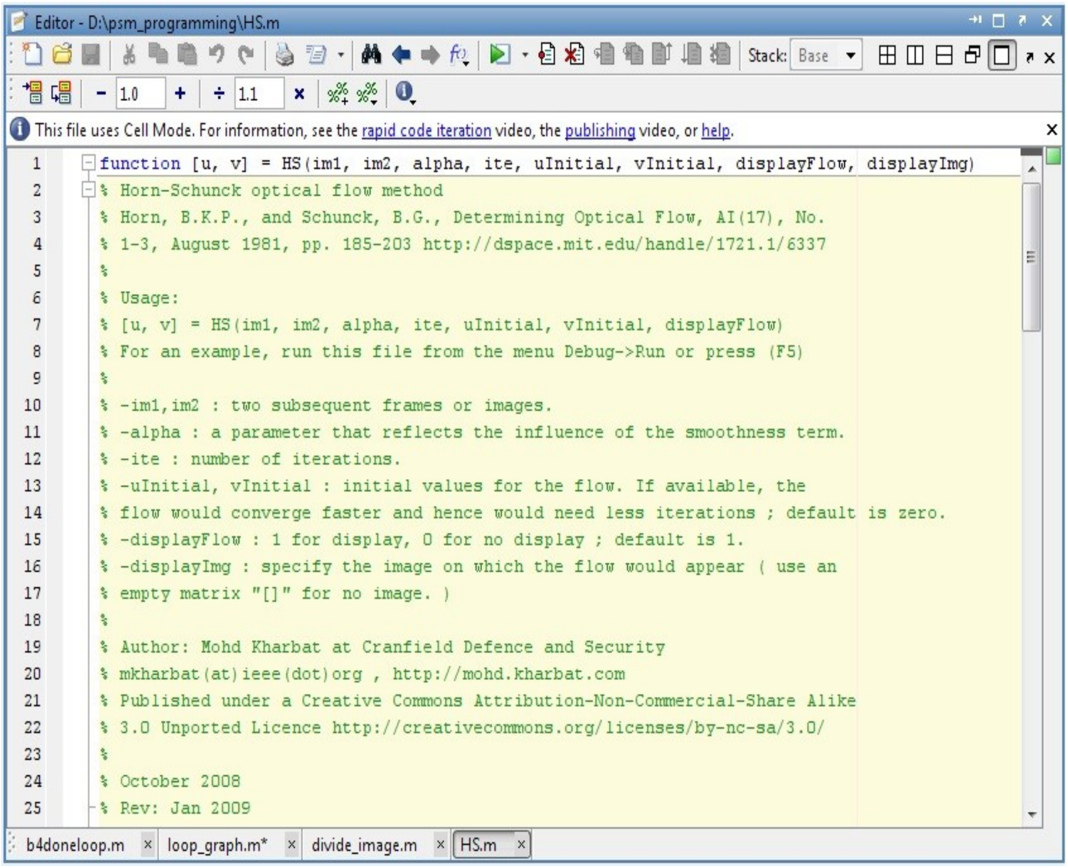



**Figure 4.2:** Horn-Schunck function description

From the function of Horn-Schunck method, we can see clearly that the two image will be analyze and the value of the variable which are *u* and *v* will be produce. The Horn-Schunck constraint defines a straight line in the *uv* velocity space.

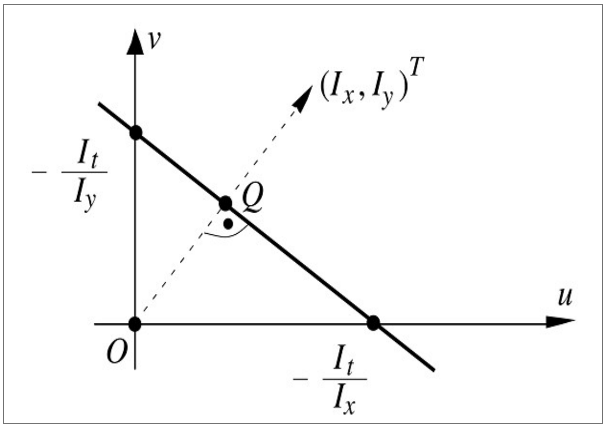



**Figure 4.3:** *uv* velocity space

The optical flow in the image plane and the velocities in the three dimensional world is not necessarily obvious. The motion were perceive by a changing picture is projected onto stationary screen [4]. So, in order to detect the moving object, the brightness will be considered. The variations in brightness due to shading effect can be evade by make an assumptions that the objects have flat surface.

Besides, the illumination across the surface is assumed to be uniform. The brightness at a point in the image is then proportional to the reflectance of the surface at the corresponding point on the object. Initially, that reflectance was assume varies smoothly and has no spatial discontinuities. This condition assures that the image brightness is differentiable.[4]

Horn and Schunck derive an equation that relates the change in image brightness at a point to the motion of brightness pattern. Let the image brghtness at the point *(x, y)* in the image plane at the time, t be denoted by *E(x, y, t).* Now consider what happens when the pattern moves, The brightness of a particular point in the pattern is constant, so that

$$\frac{dE}{dt}=0 \quad (4.1)$$

Using the Chain Rule for differentiation,

$$\frac{\partial E}{\partial x}\frac{dx}{dt}+\frac{\partial E}{\partial y}\frac{dt}{dt}+\frac{\partial E}{dt}=0 \tag{4.2}$$

Let,

$$u=\frac{dx}{dt} \text{ and } v=\frac{dy}{dt} \tag{4.3}$$

Then, one signal linear equation can get which has two unknown parameters *u* and *v*.

$$E_x u+E_y v+E_t=0 \tag{4.4}$$

an additional constraints are needed. [4]

In this case neighboring points on the object have similar velocities and the velocity field of the brightness patterns in the image varies smoothly almost everywhere. Discontinuities in flow can be expected where one object occludes another. [4]

In order to express the additional constraint is to limit the difference between the flow velocity at a point and the average velocity over a small neighborhood containing the point. Equivalently we can minimize the sum of the squares of the Laplacians of the *x* and *y* components of the flow. The Laplacians of *u* and *v* are defined as

$$\nabla^2 u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2} \text{ and } \nabla^2 v=\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2} \tag{4.5}$$

In simple situation, both Laplacians are zero. [4]

The derivatives of brightness was estimate from the discrete set of image brightness measurements available. The estimates of $E_x$, $E_y$ and $E_t$ needed to be consistent which need to be refer to the same point in the image at the same time. Horn and Schunck use a set which gives estimates of $(E_x, E_y, E_t)$ at a point in the center of a cube formed by eight measurement. Each of the estimates is the average of four first differences taken over adjacent measurements in the cube.

$$E_x \approx \frac{1}{4}\{E_{i,j+1,k}-E_{i,j,k}+E_{i+1,j+1,k}-E_{i+1,j,k}$$

$$+E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1} \}$$

$$E_y \approx \frac{1}{4} \{ E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k}$$

$$+E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1} \}$$

$$E_t \approx \frac{1}{4} \{ E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k}$$

$$+E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1} \} \qquad (4.6)$$

The unit of the length in the reference is the grid spacing interval in each image frame and the unit of time is the image sampling period. The relationship in space and time between these measurements is shown as

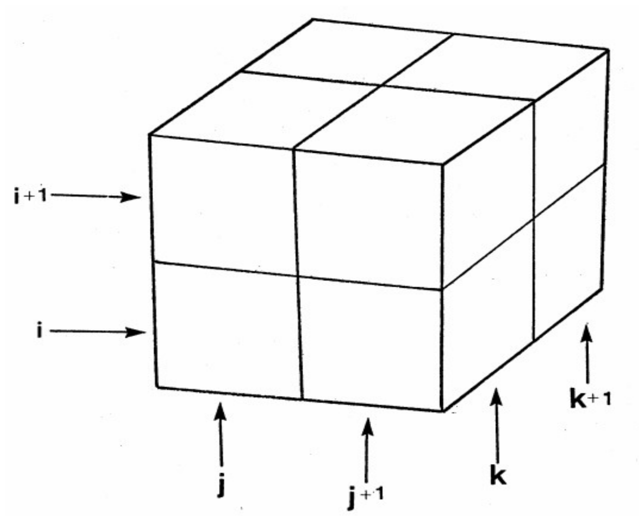



**Figure 4.4:** The three partial derivatives of image brightness.

Figure 4.4 show the partial derivatives of the image brightness at the center of the cube are each estimated from the average of first differences along four parallel edges of the cube. Here the column index *j* corresponds to the *x* direction in the image, the row index *i* to the *y* direction, while *k* lies in the time direction.

The Laplacians of *u* and *v* so should be approximated.

$$\nabla^2 u \approx k\,(\vec{u}_{i,j,k} - u_{i,j,k}) \text{ And } \nabla^2 v \approx k\,(\vec{v}_{i,j,k} - v_{i,j,k}) \qquad (4.7)$$

The proportionality factor *k* here equals 3, where the local averages *u* and *v* are defined as follows

$$\vec{u}_{i,j,k}=\frac{1}{6}\{u_{i-1,j,k}+u_{i,j+1,k}+u_{i+1,j,k}+u_{i,j-1,k}\}$$

$$+\frac{1}{12}\{u_{i-1,j-1,k}+u_{i-1,j+1,k}+u_{i+1,j+1,k}+u_{i+1,j-1,k}\}$$

$$\vec{v}_{i,j,k}=\frac{1}{6}\{v_{i-1,j,k}+v_{i,j+1,k}+v_{i+1,j,k}+v_{i,j-1,k}\}$$

$$+\frac{1}{12}\{v_{i-1,j-1,k}+v_{i-1,j+1,k}+v_{i+1,j+1,k}+v_{i+1,j-1,k}\} \quad (4.8)$$

| | | |
|---|---|---|
| 1/12 | 1/6 | 1/12 |
| 1/6 | -1 | 1/6 |
| 1/12 | 1/6 | 1/12 |

**Figure 4.5:** The Laplacian estimation

In figure 4.5, it show that the Laplacian is estimated by subtractingthe value at a point from a weighted average of the values at neighboring points. The sum of the total errors is minimized

$$\varepsilon^2=a^2\,\varepsilon_c^{\ 2}+\varepsilon_b^{\ 2} \quad (4.9)$$

From the equation above, $a^2$ is a weighting factor. Quantization error and noise may cause corruption for image brightness measurements. So, the value if $\varepsilon_b$ cannot be expect to be identically zero. This quantity will tend to have an error magnitude that is proportional to the noise in the measurement. This give the reason why $a^2$ is chosen.

Where

$$\varepsilon^2 = E_x u + E_y v \tag{4.10}$$

$$\varepsilon_c{}^2 = (\vec{u} - u)^2 + (\vec{v} - v)^2 \tag{4.11}$$

iterative solution

$$u^{n+1} = \overline{u}^n - \frac{E_x [E_x \overline{u}^n + E_y \overline{v}^n + E_t]}{a^2 + E_x{}^2 + E_y{}^2} \tag{4.12}$$

$$v^{n+1} = \overline{v}^n - \frac{E_y [E_x \overline{u}^n + E_y \overline{v}^n + E_t]}{a^2 + E_x{}^2 + E_y{}^2} \tag{4.13}$$

All of those calculation were done by MATLAB itself. The value of output from Horn-Schunck which are value *u* and *v* use to analyze the velocity of the motion occur.

# CHAPTER 5

# RESULT AND DISCUSSION

## 5.1 Introduction

In this project, programming was done in order to analysis the people motion. Before started the real motion analysis, a small analysis were done to get initial idea and some assumptions were made. This chapter, all of result of this project analysis will be stated and discuss about the overall result from this project as well this chapter related with chapter 3 where all result from chapter 3 will be shown here.

## 5.2 Initial analysis

As mention in previous chapter, which is in chapter 3, this project was started with analyze how was Horn-Schunck method work. The analysis will use the basic shapes which are circle and rectangle.

After run the Horn-Schunck method m-file, the image shows the magnitude of the motion by using arrow produce on the output image. This optical flow method was differentiating two difference image and due to the programming written in Horn-Schunck method programming, the arrow of velocity will be produce on the first image. This can be seen from the figure 5.1 and figure 5.2.

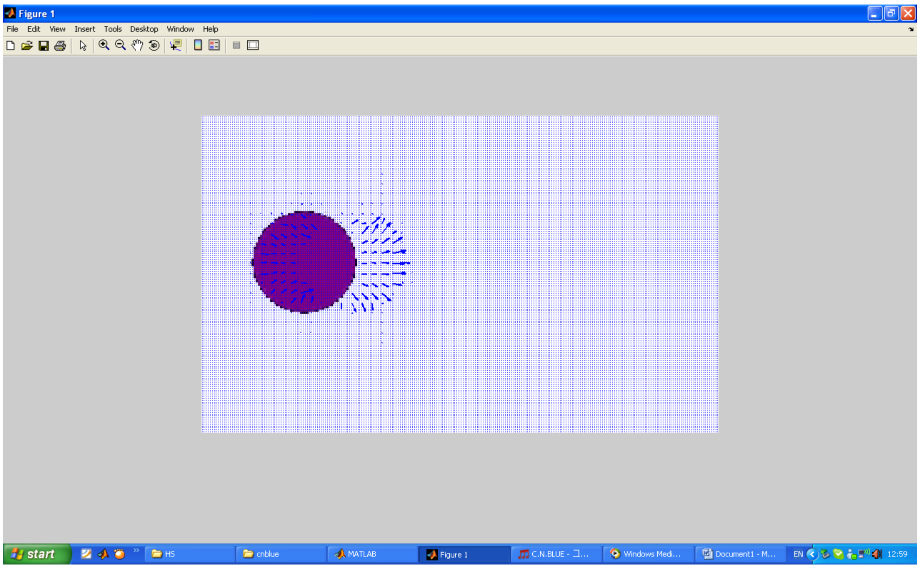


**Figure 5.1:** Arrow as output produce from two difference position of circle image

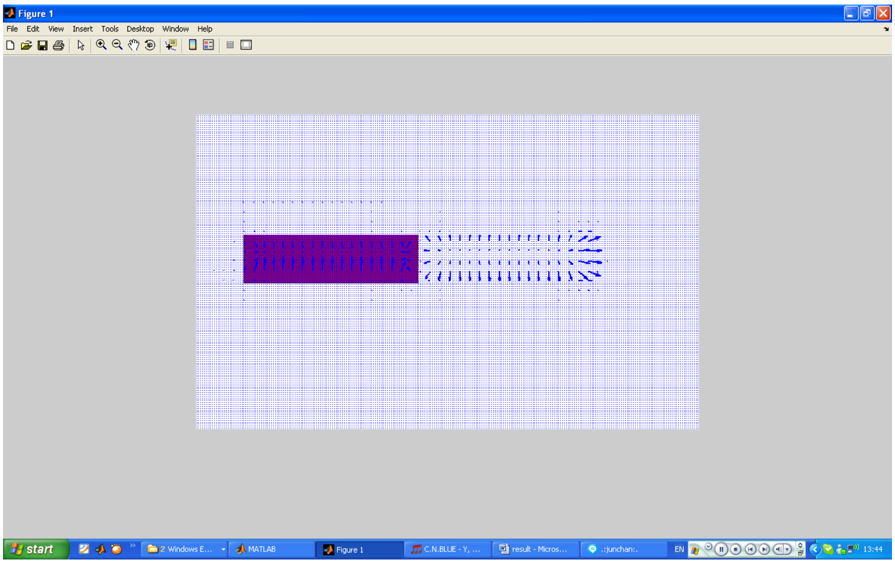


**Figure 5.2:** Arrow as output from two difference position of rectangle image

Next, assume the person movement by using ball movement that created using paint. The small distance and long distance ball of ball movement image sequence shown in figure 5.3 and figure 5.4. Below is the assumption making for every movement.

| Real situation | Assumption |
|---|---|
| Walking | Small distance of ball movement |
| Running | Long distance of ball movement |

**Table 5.1:** Assumption that related to real situation

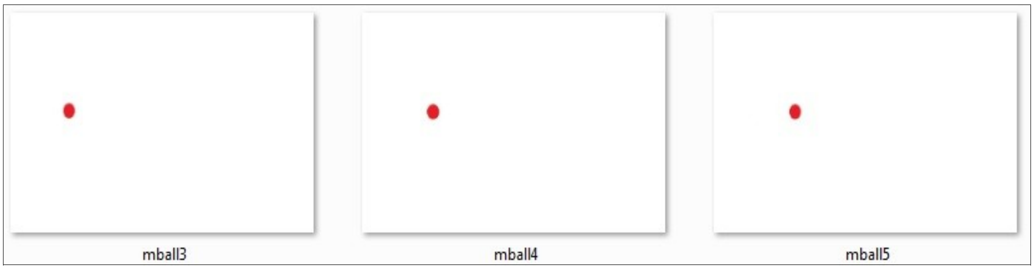



**Figure 5.3:** Image sequence for small distance ball

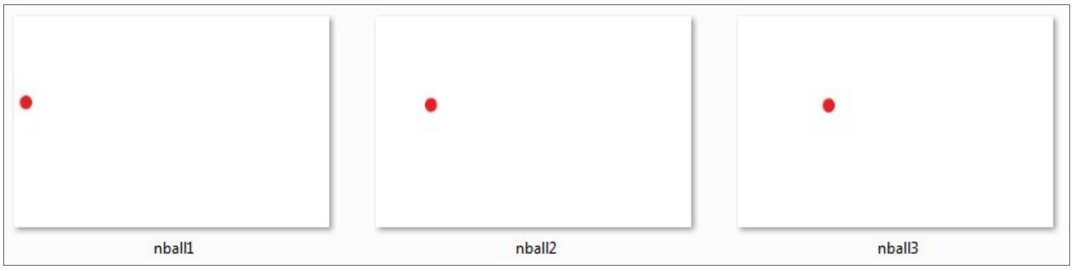



**Figure 5.4:** Image sequence for long distance ball

Then, this project proceed by observe the small distance and long distance ball of ball movement, as well as decide which is the suitable parameter to analyze the output. From figure 5.5 until figure 5.9 show that the value of minimum, maximum, mean, variance and standard deviation value of velocity that collected and printed in notepad for small distance of ball movement for a image sequence contained 6 image in a sequent. While, in figure 5.10 until figure 5.14 are for long distance of ball movement image sequence.

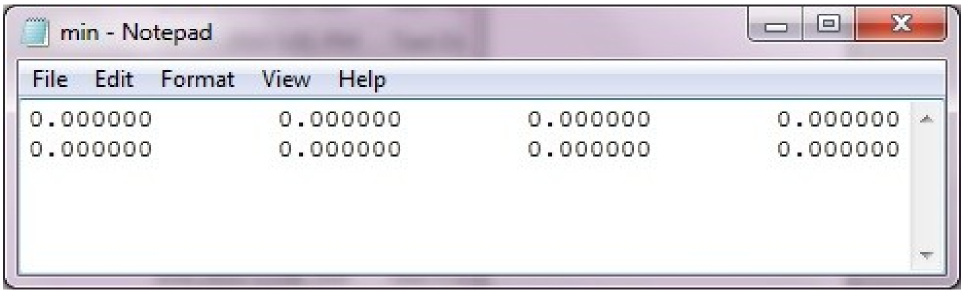
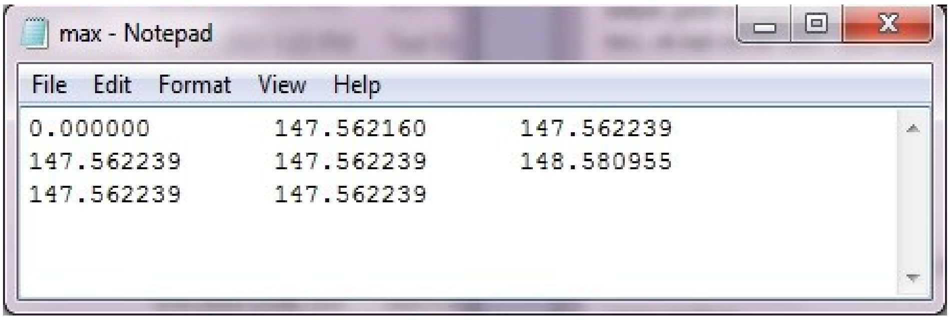
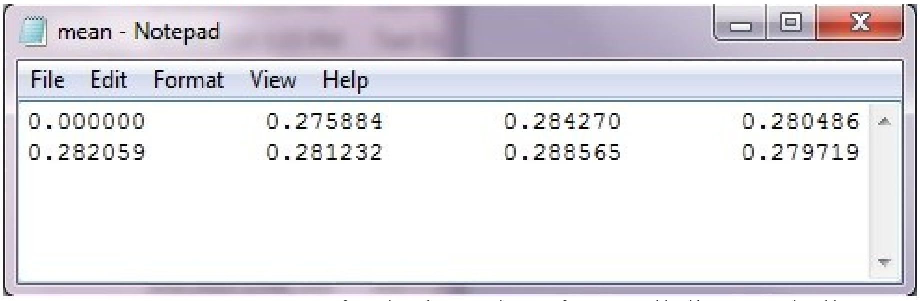



**Figure 5.5:** Minimum value of velocity for small distance ball



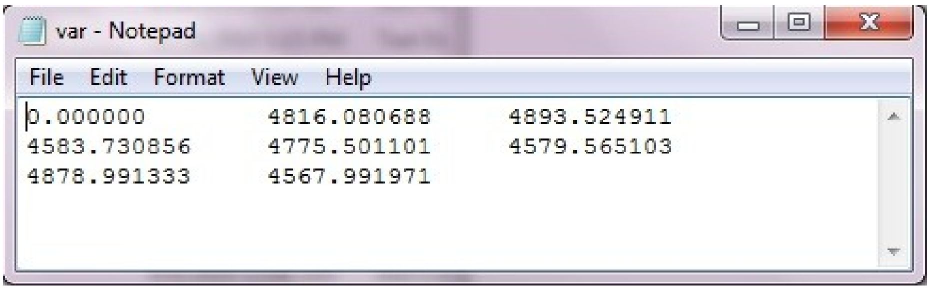

| | | |
|---|---|---|
| 0.000000 | 147.562160 | 147.562239 |
| 147.562239 | 147.562239 | 148.580955 |
| 147.562239 | 147.562239 | |

**Figure 5.6:** Maximum value of velocity for small distance ball



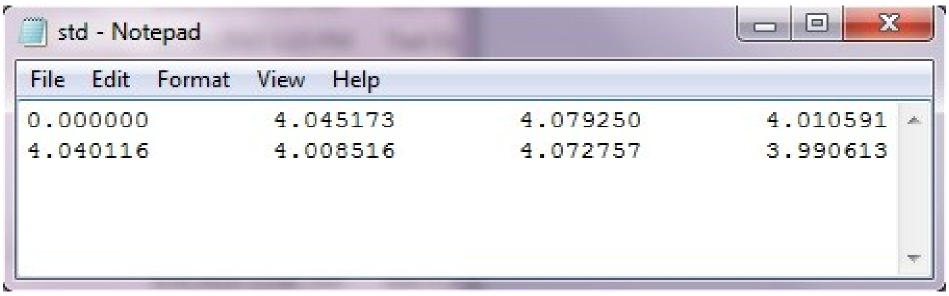

| | | | |
|---|---|---|---|
| 0.000000 | 0.275884 | 0.284270 | 0.280486 |
| 0.282059 | 0.281232 | 0.288565 | 0.279719 |

**Figure 5.7:** Mean of velocity values for small distance ball




| | | |
|---|---|---|
| 0.000000 | 4816.080688 | 4893.524911 |
| 4583.730856 | 4775.501101 | 4579.565103 |
| 4878.991333 | 4567.991971 | |

**Figure 5.8:** Variance of velocity values for small distance ball




| | | | |
|---|---|---|---|
| 0.000000 | 4.045173 | 4.079250 | 4.010591 |
| 4.040116 | 4.008516 | 4.072757 | 3.990613 |

**Figure 5.9:** Standard deviation of velocity values for small distance ball

min - Notepad

| | | | |
|---|---|---|---|
| 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.000000 | 0.000000 | 0.000000 | 0.000000 |

**Figure 5.10:** Minimum value of velocity for long distance ball

max - Notepad

| | | |
|---|---|---|
| 0.000000 | 131.269026 | 151.124185 |
| 117.383943 | 118.562662 | 128.165383 |
| 117.383943 | 137.957770 | |

**Figure 5.11:** Maximum value of velocity for long distance ball

mean - Notepad

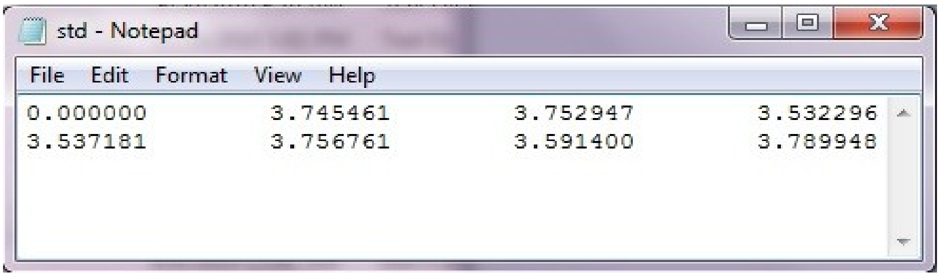
| | | | |
|---|---|---|---|
| 0.000000 | 0.248580 | 0.249896 | 0.234798 |
| 0.238479 | 0.257740 | 0.241857 | 0.252369 |

**Figure 5.12:** Mean of velocity values for long distance ball

var - Notepad

| | | |
|---|---|---|
| 0.000000 | 3554.644085 | 3622.021344 |
| 2676.409639 | 2783.796611 | 3486.613343 |
| 2865.668337 | 3588.619759 | |

**Figure 5.13:** Variance of velocity values for long distance ball



std - Notepad

File Edit Format View Help

| | | | |
|---|---|---|---|
| 0.000000 | 3.745461 | 3.752947 | 3.532296 |
| 3.537181 | 3.756761 | 3.591400 | 3.789948 |

**Figure 5.14:** Standard deviation of velocity values for long distance ball

After mean has been decide for further analysis, it values plotted in the graph to facilitate observation. Mean of velocity value for small distance and long distance were plotted in a graph as shown in figure 5.15 and figure 5.16. From the value that plotted on both graph, show that there are differences. The long distance ball produce mean value lower than small distance ball.

From figure 5.15, the mean value that produce by small distance of moving ball which is shown in figure 5.3 are greater than 0.25. While in figure 5.16 the mean value that produce by image sequences shown in figure 5.4 are less than 0.25. This output of graph will give the result of the initial analysis as well as give the idea for real analysis. So that, the running image sequence will produce the lower mean value for each sequent than walking image sequence.

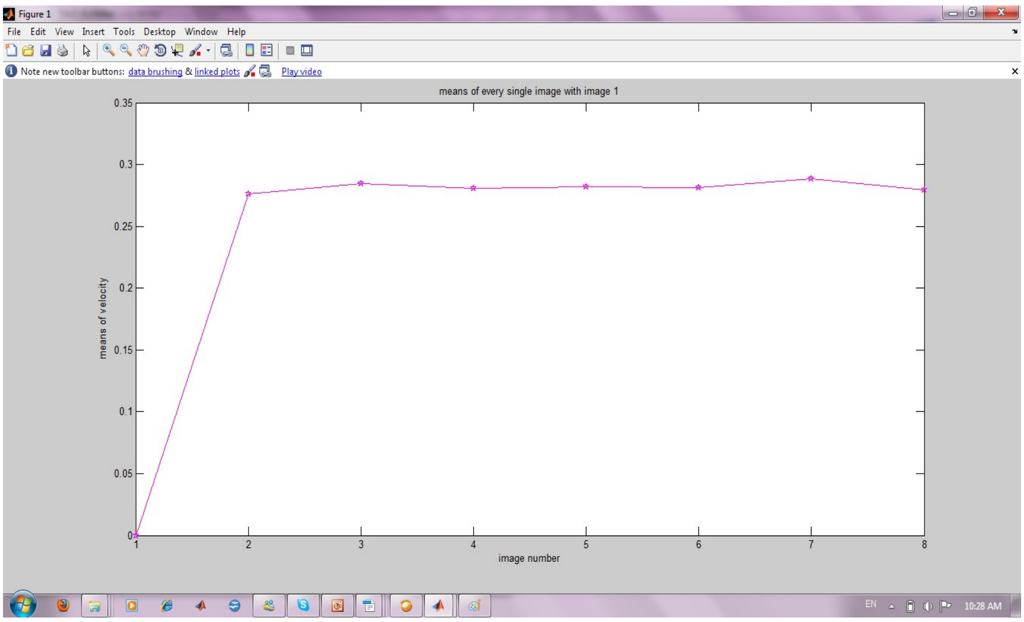



**Figure 5.15:** Graph for small distance of moving ball

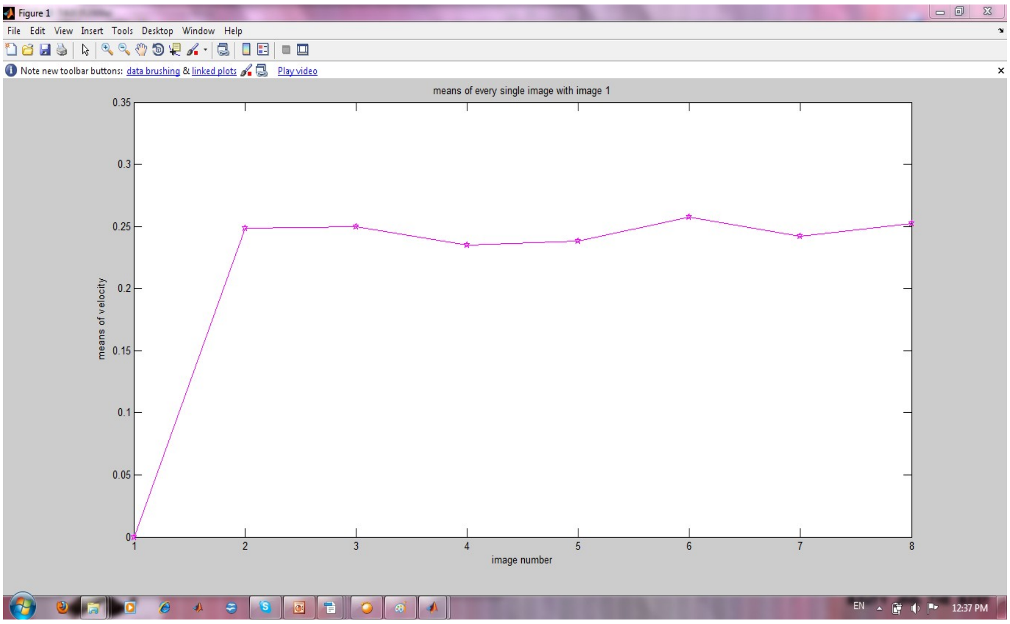



**Figure 5.16:** Graph for long distance of moving ball

## 5.3 Further analysis

Further analysis or known as real analysis used real movement a person. While doing this analysis, found that there are background motion such as movement of flag and tree. A solution was finding in order to eliminate the value of background motion that effected the mean value of main motion in the image sequence.

A method to reduce or eliminate noise or the unnecessary motion is to do sub-block of velocity for whole image into sixteen (16) sub-blocks. Figure 5.17 and figure 5.18 show how was the velocity value divided into 16 sub-blocks. It represented as image because Horn-Schunck method represents velocity based on number of pixel.

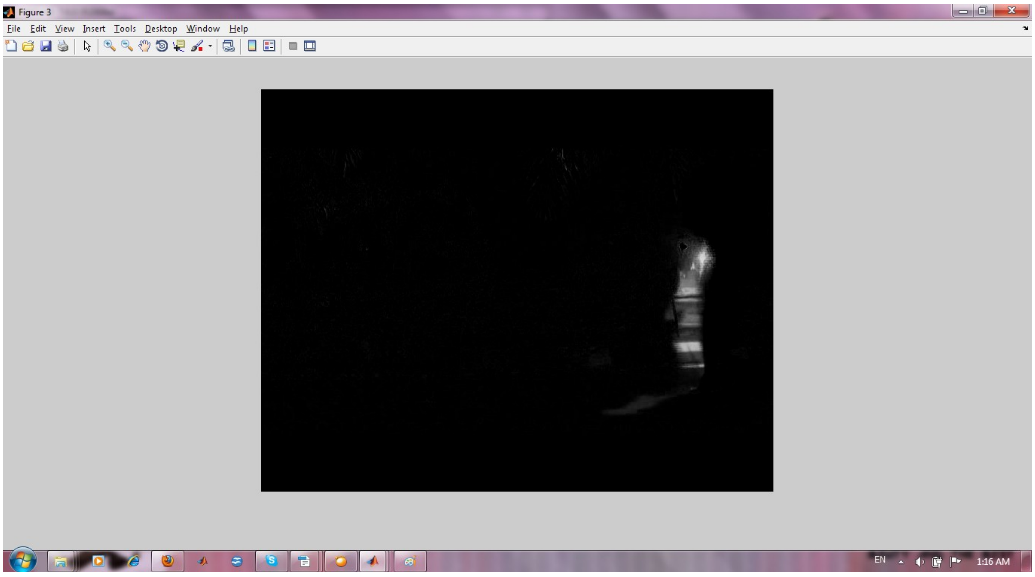



**Figure 5.17:** Image before sub-block

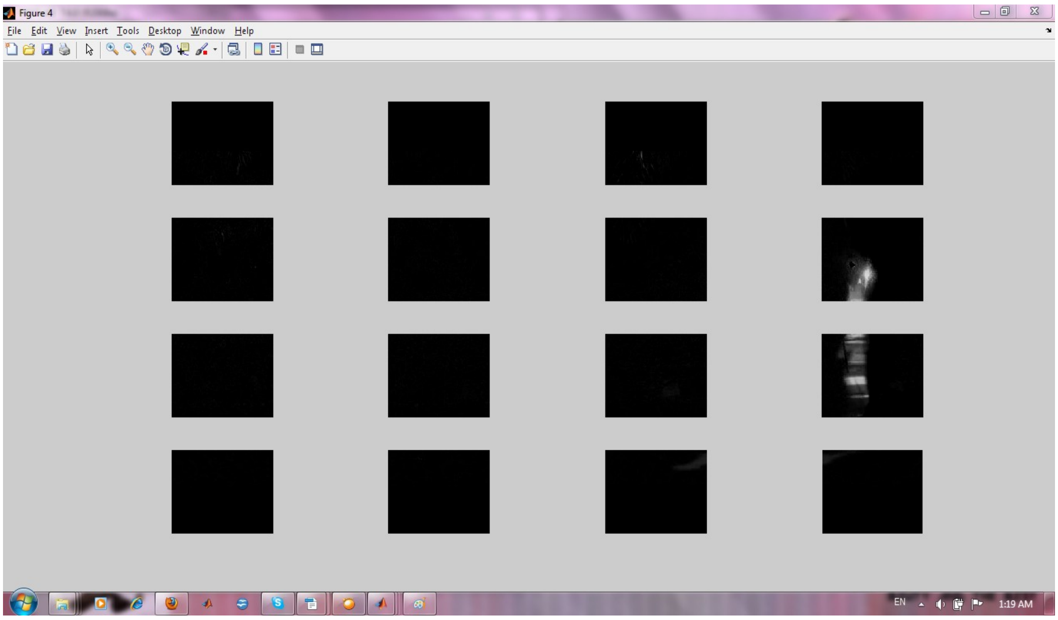
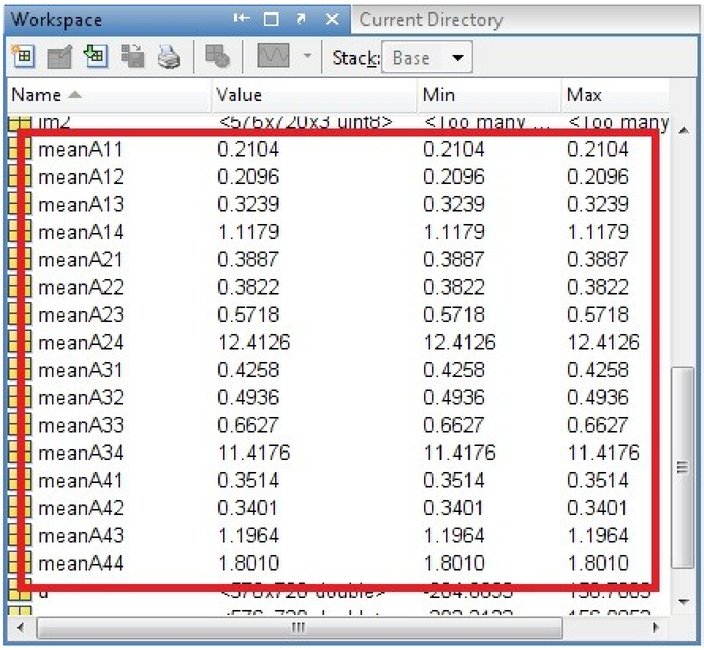



**Figure 5.18:** Image after sub-block

Each sub-block name represent as in figure 5.19 and in figure 5.20 show the mean value for each sub-block before do eliminated of mean for background noise.

| | | | |
|---|---|---|---|
| A11 | A12 | A13 | A14 |
| A21 | A22 | A23 | A24 |
| A31 | A32 | A33 | A34 |
| A41 | A42 | A43 | A44 |

**Figure 5.19:** Sub-block of velocity



| Name | Value | Min | Max |
|---|---|---|---|
| im2 | <576x720x3 uint8> | <Too many ... | <Too many |
| meanA11 | 0.2104 | 0.2104 | 0.2104 |
| meanA12 | 0.2096 | 0.2096 | 0.2096 |
| meanA13 | 0.3239 | 0.3239 | 0.3239 |
| meanA14 | 1.1179 | 1.1179 | 1.1179 |
| meanA21 | 0.3887 | 0.3887 | 0.3887 |
| meanA22 | 0.3822 | 0.3822 | 0.3822 |
| meanA23 | 0.5718 | 0.5718 | 0.5718 |
| meanA24 | 12.4126 | 12.4126 | 12.4126 |
| meanA31 | 0.4258 | 0.4258 | 0.4258 |
| meanA32 | 0.4936 | 0.4936 | 0.4936 |
| meanA33 | 0.6627 | 0.6627 | 0.6627 |
| meanA34 | 11.4176 | 11.4176 | 11.4176 |
| meanA41 | 0.3514 | 0.3514 | 0.3514 |
| meanA42 | 0.3401 | 0.3401 | 0.3401 |
| meanA43 | 1.1964 | 1.1964 | 1.1964 |
| meanA44 | 1.8010 | 1.8010 | 1.8010 |

**Figure 5.20:** Each sub-block mean value before background mean elimination

Based on figure 5.18 until figure 5.20 the mean value for each sub-block can be understood by observe their relation. Hence, mean value clearly show that for sub-block A24 and A34 there is motion and the value of mean greater than other sub-block. So, the 2 is threshold value to eliminate the background. If the mean value below than 2, the velocity for that sub-block will be set as zero (0).

After set the threshold value, then the mean value produce for each sub-block as shown in figure 5.21. Both, mean in A24 and A34 sub-block left after the elimination. It is because clearly shown there is no other motion beside both sub-block. Later, the summation of the value of mean for each block, in order to get the mean value for whole block. For instance, from figure 5.21, the summation of mean value only for  A24 and A34 sub-block that will produce 23.8302 as the value of mean for a  image sequent.

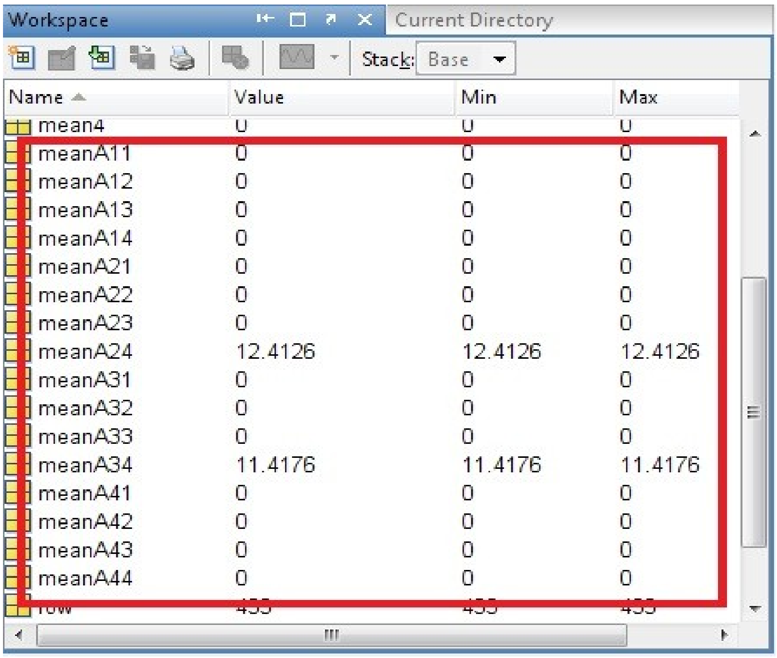


**Figure 5.21:** Mean value for each sub-block after mean noise elimination

Afterward, from the mean for whole block, will be analyze to get threshold value to do classification either the image sequence represent running or walking. Later, after do image sequence indicator in this system, the output of this system will come out with how much running or walking image were indicate before produce the output that tell either the images sequence is running or walking images sequence. Its show on the command window. Figure 5.22 and figure 5.23 shows how was the output stated on the command window.

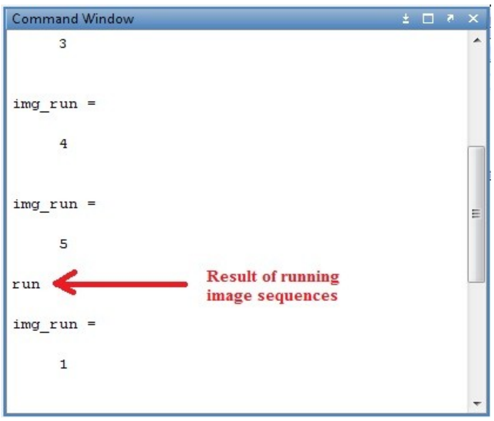



**Figure 5.22:** Result for running image sequences

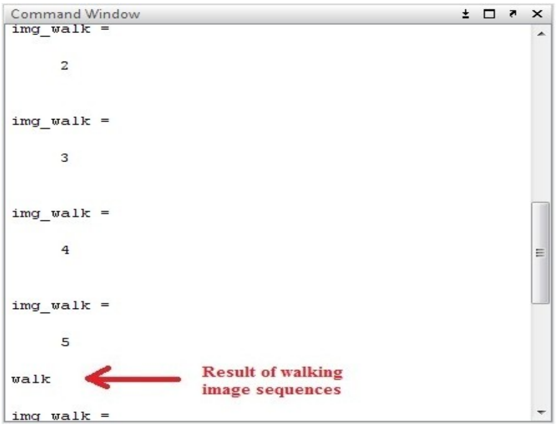



**Figure 5.23:** Result for walking image sequences

Hence, this system can detect and differentiate the normal and abnormal motion through an image sequence.

Efficiency calculation needed in order to know how much efficient this system. As mention before in chapter 3, the image sequent indicator will be modified to get the best indicator conditions for this system. Table 5.2 show the percentage for each condition for each difference image sequence.

Figure 5.24 show the output of running images sequence while figure 5.25 show the output of walking images sequences. In order to calculate the system efficiency if in the image sequence, the output will be not the correct output will be consider as error and this shown in figure 5.26.

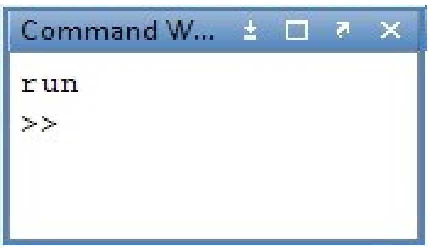



**Figure 5.24:** Output of running images sequence

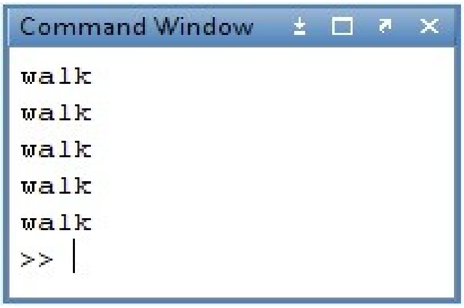



**Figure 5.25:** Output of walking images sequence

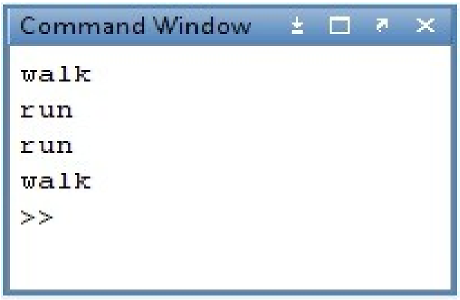



**Figure 5.26:** Error output in images sequence

This efficiency of image sequent indicator will be test on 4 image sequences for walking where each sequence from 4 difference person as well as running image sequences. This image sequence will be test and the output from this system will be analyzed. So that, the efficient image sequent indicator can be selected in order to make this motion detection system more efficiency while producing the output.

| | run | walk | Percentage of efficiency, % |
|---|---|---|---|
| **Indicator condition** | 5 | 5 | 0, a sequence can be detect |
| | 5 | 3 | 0, a sequence can be detect |
| | 3 | 5 | 81 |
| | 3 | 3 | 89 |

**Table 5.2:** System efficiency by indicator conditions

At the end of this motion analysis system, 5 images sequent indicator of run and 5 images sequent indicator of walk produce no efficiency for this system it is because out of 4 running image sequences, there is one image sequence cannot be detect either the sequence is running or walking image sequence. Same thing happen while setup 5 images sequent of run and 3 images sequence of walk.

Setting up 3 images sequent indicator of run and 5 images sequent indicator of walk make this system had 81% of efficient. It is because while testing walking image sequence, there are 3 sequence of walking image produce run condition as their output beside producing walking image sequence output.

After that, by setting up 3 images sequent indicator of run and 3 images sequent indicator of walk make this system had 89% of efficient. This happen because of while testing running image sequence, there are a sequence of walking and running image produce run and walk condition as their output beside producing walking image sequence output.

The system efficiency testing make this system 89% efficient by using 3 images sequent of run indicator and 3 images sequent of walk indicator condition in this system.

# CHAPTER 6

# CONCLUSION AND RECOMMENDATION

## 6.1 Introduction

In this project, the description of these motion detection and image processing analysis project, already been discussed accordingly based from previous chapters. This project flow and development are also discussed in more details in chapter 3 while in the chapter 4, the further discussion and final result are been focus as the important determinant to fulfill all the objective of this project.

## 6.2 Conclusion

In order to detect motion, the optical flow method had been developed based on objectives of this project. These application of optical flow are been use to detect abnormal motion based on security system. Additionally, this project also fulfill the second objective which is by applying optical flow in image processing analysis, that contribute to focusing on scope of this project.

The main scopes of this project are to implement Horn-Schunck method of the Optical flow by using MATLAB software. Based on the Horn-Schunck method calculations are needed to contribute for second scopes. Calculation are been applied to analyze the motion image.

## 6.3 Recommendations

After several improvement study and analysis in this project, there were some recommendation and further development listed below to improve this motion detection system to give better performance. The recommendations are:

i. This system only analyze 576x720 pixel of image. Hence, some changes in term of programming code of this system in order to make this system more user friendly by making various size of image can be analyze.
ii. Do more analyze about the various human movement pattern.
iii. Do analyze of motion detection online which is directly from video or closed-circuit television (CCTV).

# REFERENCES

[1] Shuo Liu. (2009). “Object Trajectory Estimation Using Optical Flow”. Utah State University. Bachelor Thesis.

[2] The MathWorks, Inc. (1996). "MATLAB The Language of Technical Computing", Language Reference Manual, Copyright 1984 – 1996. All Rights Reserved.

[3] Weiqi Yuan and Binxiu Gao (March 2006). "Iris Recognition System Based on Difference of Two Images Gray", ICGST International Journal on Graphics, Vision and Image Processing (GVIP) Special Issue on Object Tracking and Recognition.

[4] Dr A D Marshall, “Vision System”. Cardiff University, Copyright 1994-1997. Teaching Note. Available : http://www.cs.cf.ac.uk/Dave/Vision_lecture/Vision_lecture_caller.html

[5] Mark S. Nixon and Alberto S, Aguad. (2004). “Feature extraction and image processing”. The Computer Journal, Vol 47(2), page 271.

[6] Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart (2003), "Image Processing Operator Worksheets". Hypermedia Image Processing Reference, Copyright 2000. All Rights Reserved.
Available : http://homepages.inf.ed.ac.uk/rbf/HIPR2/classify.htm

[7] Te-Won Lee and Michael S. Lewicki (March 2002) "Unsupervised image classification. Segmentation and enhancement" IEEE Transaction on Image Processing, Vol 11, No 3, pages 270 - 279

[8] A John Maule. (2010). "Can computer help ovecone limitations in human decision making?" International Journal of Human-Computer Interation, Vol 26, Issue 2 and 3, pages 108-109.

[9] Brandon C.Welsh, Mark E.Mudge and David P.Farrington. (16 March 2009). "Reconceptualizing public area surveillance and crime prevention: Security guards, place managers and defensible space", Security Journal advance

online publication.

[10] Timothy Coupe and Satverg Kaur. (April 2005) "The Role of Alarms and CCTV in Detecting Non-residential Burglary", Security Journal 18, 53-72.

[11] Ce Liu, William T. Freeman, Edward H. Adelson, Yair Weiss. (2008). "Human-Assisted Motion Annotation" , IEEE Conference, Computer Vision and Pattern Recognition. Research Report.

[12] Kui Liu, Qian Du, He Yang, and BenMa. (16 January 2010). "Optical Flow and Principal Component Analysis-Based Motion Detection in Outdoor Videos", EURASIP Journal on Advances in Signal Processing, Volume 2010, Article ID 680623, 6 pages.

[13] Jacky Baltes. (2003) "Optical Flow Algorithms" Teaching Note, University of Manitoba, Winnipeg, Canada. Teaching Note. Available : http://www4.cs.umanitoba.ca/~jacky/Teaching/Courses/74.795-LocalVision/2005/Lectures/

[14] Yana Momchilova Mileva. (2009). "Invariance with Optic Flow", Mathematical Image Analysis Group, Department of Computer Science, Saarland University. Master Thesis.

[15] Deqing Sun,StefanRoth, J.P. Lewis,and Michael J.Black. (2008). “Learning optical flow”. Springer-Verlag Berlin Heidelberg 2008. In D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 83–97.

[16] B. K. P. Horn and B. Schunck. (1993). “Determining optical flow: A retrospective”, Artificial Intelligence, vol. 59, pp. 81–87. Available: http://www.sciencedirect.com/science/article/B6TYF-4811SSS-1Y/2/2ecf1379d9da8d5dc3b398e044b5a24e

[17] B. K. P. Horn and B. G. Schunck. (1981). “Determining optical flow,” Artificial Intelligence, vol. 17, no. 1–3, pp. 185–203.

[18] Roth, S., Black, M.J. (2007). "On the spatial statistics of optical flow" International Journal of Computer Vision (IJCV 2007), Vol 74, pp. 33–50.

[19] Brox, T., Bruhn, A., Papenberg, N., Weickert. J. (2003). "High accuracy optical flow estimation based on a theory for warping". In: Pajdla, T., Matas, J(G.) (eds.) European Conference on Computer Vision (ECCV) 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg.

[20] A. Bruhn, J. Weickert, and C. Schnorr. (2005). “Lucas/Kanade meets

Horn/Schunck: combining local and global optic flow methods,” International Journal of Computer Vision, vol. 61, no. 3, pp. 211–231.

[21] S.S. Beauchemin and J.L. Barron. (September 1995). "The Computation of Optical Flow". ACM Computing Survey, Vol 27(3), pp 4.33-467.

[22] J.L. Barron, D.J. Fleet and S S Beauchemin. (1994). "Performance of Optical FlowTechniques", International Journal of Computer Vision (IJCV1994), Vol 12(1), pp. 43-77.

[23] B. Galvin, B. McCane, K. Novins, D. Mason and S. Mills. (1998). "Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms", British Machine Vision Conference, 1998

# APPENDIX I
# HORN-SCHUNCK METHOD PROGRAMMING CODE
(MAIN PROGRAMMING CODE)

```
function [u, v] = HS(im1, im2, alpha, ite, uInitial, vInitial,
displayFlow, displayImg)
% Horn-Schunck optical flow method
% Horn, B.K.P., and Schunck, B.G., Determining Optical Flow, AI(17),
No.
% 1-3, August 1981, pp. 185-203
http://dspace.mit.edu/handle/1721.1/6337
%
% Usage:
% [u, v] = HS(im1, im2, alpha, ite, uInitial, vInitial, displayFlow)
% For an example, run this file from the menu Debug->Run or press
(F5)
%
% -im1,im2 : two subsequent frames or images.
% -alpha : a parameter that reflects the influence of the smoothness
term.
% -ite : number of iterations.
% -uInitial, vInitial : initial values for the flow. If available,
the
% flow would converge faster and hence would need less iterations ;
default is zero.
% -displayFlow : 1 for display, 0 for no display ; default is 1.
% -displayImg : specify the image on which the flow would appear
( use an
% empty matrix "[]" for no image. )
%
% Author: Mohd Kharbat at Cranfield Defence and Security
% mkharbat(at)ieee(dot)org , http://mohd.kharbat.com
% Published under a Creative Commons Attribution-Non-Commercial-
Share Alike
% 3.0 Unported Licence http://creativecommons.org/licenses/by-nc-
sa/3.0/
%
% October 2008
% Rev: Jan 2009

%% Default parameters

%image

if nargin<1 || nargin<2
    im1=imread('D:\psm_data
collection\output\walking\walk\walk_4.jpg');
    im2=imread('D:\psm_data
collection\output\walking\walk\walk_5.jpg');
end

%

if nargin<3
    alpha=1;
end
if nargin<4
    ite=100;
end
if nargin<5 || nargin<6
    uInitial = zeros(size(im1(:,:,1)));
    vInitial = zeros(size(im2(:,:,1)));
elseif size(uInitial,1) ==0 || size(vInitial,1)==0
    uInitial = zeros(size(im1(:,:,1)));
    vInitial = zeros(size(im2(:,:,1)));
```

```
end
if nargin<7
    displayFlow=0;
end
if nargin<8
    displayImg=im1;
end

%% Convert images to grayscale


%image
if size(size(im1),2)==3
    im1=rgb2gray(im1);
end
if size(size(im2),2)==3
    im2=rgb2gray(im2);
end
im1=double(im1);
im2=double(im2);

im1=smoothImg(im1,1);
im2=smoothImg(im2,1);

tic;

%% Set initial value for the flow vectors

u = uInitial;
v = vInitial;

% Estimate spatiotemporal derivatives
[fx, fy, ft] = computeDerivatives(im1, im2);

% Averaging kernel
kernel_1=[1/12 1/6 1/12;1/6 0 1/6;1/12 1/6 1/12];

% Iterations
for i=1:ite
    % Compute local averages of the flow vectors
    uAvg=conv2(u,kernel_1,'same');
    vAvg=conv2(v,kernel_1,'same');
    % Compute flow vectors constrained by its local average and the
optical flow constraints
    u= uAvg - ( fx .* ( ( fx .* uAvg ) + ( fy .* vAvg ) + ft ) ) ./
( alpha^2 + fx.^2 + fy.^2);
    v= vAvg - ( fy .* ( ( fx .* uAvg ) + ( fy .* vAvg ) + ft ) ) ./
( alpha^2 + fx.^2 + fy.^2);
end

u(isnan(u))=0;
v(isnan(v))=0;

%% Plotting
if displayFlow==1
    plotFlow(u, v, displayImg, 5, 5);
end
```

# APPENDIX II
# HORN-SCHUNCK METHOD PROGRAMMING CODE
(PLOTFLOW PROGRAMMING CODE)

```
function plotFlow(u, v, imgOriginal, rSize, scale)
% Creates a quiver plot that displays the optical flow vectors on
the
% original first frame (if provided). See the MATLAB Function
Reference for
% "quiver" for more info.
%
% Usage:
% plotFlow(u, v, imgOriginal, rSize, scale)
%
% u and v are the horizontal and vertical optical flow vectors,
% respectively. imgOriginal, if supplied, is the first frame on
which the
% flow vectors would be plotted. use an empty matrix '[]' for no
image.
% rSize is the size of the region in which one vector is visible.
scale
% over-rules the auto scaling.
%
% Author: Mohd Kharbat at Cranfield Defence and Security
% mkharbat(at)ieee(dot)org , http://mohd.kharbat.com
% Published under a Creative Commons Attribution-Non-Commercial-
Share Alike
% 3.0 Unported Licence http://creativecommons.org/licenses/by-nc-
sa/3.0/
%
% October 2008
% Rev: Jan 2009

figure();

if nargin>2
    if sum(sum(imgOriginal))~=0
        imshow(imgOriginal,[0 255]);
        hold on;
    end
end
if nargin<4
    rSize=5;
end
if nargin<5
    scale=3;
end

% Enhance the quiver plot visually by showing one vector per region
for i=1:size(u,1)
    for j=1:size(u,2)
        if floor(i/rSize)~=i/rSize || floor(j/rSize)~=j/rSize
            u(i,j)=0;
            v(i,j)=0;
        end
    end
end
quiver(u, v, scale, 'color', 'b', 'linewidth', 2);
set(gca,'YDir','reverse');
```

**APPENDIX III**
**HORN-SCHUNCK METHOD PROGRAMMING CODE**
(SMOOTHING IMAGE  PROGRAMMING CODE)

```
function smoothedImg=smoothImg(img,segma)
% Convolving an image with a Gaussian kernel.

% The degree of smoothing is determined by the Gaussian's standard
% deviation 'segma'. The Gaussian outputs a 'weighted average' of
each
% pixel's neighborhood, with the average weighted more towards the
value of
% the central pixels. The larger the standard deviation, the less
weight
% towards the central pixels and more weight towards the pixels
away, hence
% heavier blurring effect and less details in the output image.
%
% Author: Mohd Kharbat at Cranfield Defence and Security
% mkharbat(at)ieee(dot)org , http://mohd.kharbat.com
% Published under a Creative Commons Attribution-Non-Commercial-
Share Alike
% 3.0 Unported Licence http://creativecommons.org/licenses/by-nc-
sa/3.0/
%
% October 2008

if nargin<2
    segma=1;
end

G=gaussFilter(segma);
smoothedImg=conv2(img,G,'same');
smoothedImg=conv2(smoothedImg,G','same');
```

**APPENDIX IV**
**HORN-SCHUNCK METHOD PROGRAMMING CODE**
(COMPUTE DERIVATIVE  PROGRAMMING CODE)

```
function [fx, fy, ft] = computeDerivatives(im1, im2)

if size(im2,1)==0
    im2=zeros(size(im1));
end

% Horn-Schunck original method
fx = conv2(im1,0.25* [-1 1; -1 1],'same') + conv2(im2, 0.25*[-1 1;
-1 1],'same');
fy = conv2(im1, 0.25*[-1 -1; 1 1], 'same') + conv2(im2, 0.25*[-1 -1;
1 1], 'same');
ft = conv2(im1, 0.25*ones(2),'same') + conv2(im2,
-0.25*ones(2),'same');

% derivatives as in Barron
% fx= conv2(im1,(1/12)*[-1 8 0 -8 1],'same');
% fy= conv2(im1,(1/12)*[-1 8 0 -8 1]','same');
% ft = conv2(im1, 0.25*ones(2),'same') + conv2(im2,
-0.25*ones(2),'same');
% fx=-fx;fy=-fy;

% An alternative way to compute the spatiotemporal derivatives is to
use simple finite difference masks.
% fx = conv2(im1,[1 -1]);
% fy = conv2(im1,[1; -1]);
% ft= im2-im1;
```

**APPENDIX V**
**HORN-SCHUNCK METHOD PROGRAMMING CODE**
(FILTERING PROGRAMMING CODE)

```
function G=gaussFilter(segma,kSize)
% Creates a 1-D Gaussian kernel of a standard deviation 'segma' and
a size
% of 'kSize'.
%
% In theory, the Gaussian distribution is non-zero everywhere. In
practice,
% it's effectively zero at places further away from about three
standard
% deviations. Hence the reason why the kernel is suggested to be
truncated
% at that point.
%
% The 2D Gaussian filter is a complete circular symmetric operator.
It can be
% seperated into x and y components. The 2D convolution can be
performed by
% first convolving with 1D Gaussian in the x direction and the same
in the
% y direction.
%
% Author: Mohd Kharbat at Cranfield Defence and Security
% mkharbat(at)ieee(dot)org , http://mohd.kharbat.com
% Published under a Creative Commons Attribution-Non-Commercial-
Share Alike
% 3.0 Unported Licence http://creativecommons.org/licenses/by-nc-
sa/3.0/
%
% October 2008

if nargin<1
    segma=1;
end
if nargin<2
    kSize=2*(segma*3);
end

x=-(kSize/2):(1+1/kSize):(kSize/2);
G=(1/(sqrt(2*pi)*segma)) * exp (-(x.^2)/(2*segma^2));
```

**APPENDIX VI**
**ANALYSIS OF MOTION DETECTION PROGRAMMING CODE**

```
clc
clear all

imgreference=imread('D:\psm_data
collection\output\running\imagerun\b_1.jpg');

img_run=0;
img_walk=0;

% create the text
fid_means=fopen('D:\psm_programming\data_analysis\means.txt','w+');


nImage=60;
text='D:\psm_data collection\output\running\imagerun\b_';

 for im=2:nImage
     filname=strcat(text,int2str(im));
     filname=strcat(filname,'.jpg');
     img=imread(filname);

     [u v]=HS(imgreference,img);

     U=u.^2;
     V=v.^2;
     velocity=sqrt(U+V);

     for row=1:144:576
         for col=1:180:720
             if mean(mean(velocity(row:row+143, col:col+179)))<2;
                 velocity(row:row+143, col:col+179)=0;
             end
         end
     end

     meanA11=mean(mean(velocity(1:144, 1:180)));
     meanA12=mean(mean(velocity(1:144, 181:360)));
     meanA13=mean(mean(velocity(1:144, 361:540)));
     meanA14=mean(mean(velocity(1:144, 541:720)));

     meanA21=mean(mean(velocity(145:288, 1:180)));
     meanA22=mean(mean(velocity(145:288, 181:360)));
     meanA23=mean(mean(velocity(145:288, 361:540)));
     meanA24=mean(mean(velocity(145:288, 541:720)));

     meanA31=mean(mean(velocity(289:432, 1:180)));
     meanA32=mean(mean(velocity(289:432, 181:360)));
     meanA33=mean(mean(velocity(289:432, 361:540)));
     meanA34=mean(mean(velocity(289:432, 541:720)));

     meanA41=mean(mean(velocity(433:576, 1:180)));
     meanA42=mean(mean(velocity(433:576, 181:360)));
     meanA43=mean(mean(velocity(433:576, 361:540)));
     meanA44=mean(mean(velocity(433:576, 541:720)));

     mean1=meanA11+meanA12+meanA13+meanA14;
     mean2=meanA21+meanA22+meanA23+meanA24;
     mean3=meanA31+meanA32+meanA33+meanA34;
     mean4=meanA41+meanA42+meanA43+meanA44;

     allmean=mean1+mean2+mean3+mean4;
```

```
%       write the data means to text
     fprintf(fid_means,'%f',allmean );
     fprintf(fid_means,'\t');


%       making decision
     tresh_mean=(allmean>18);

     if tresh_mean<1
         img_run=img_run+1
         img_walk=0;
         if img_run==5
             disp('run')
             img_run=0;
             img_walk=0;
         else
         end
     elseif tresh_mean>0
         img_walk=img_walk+1
         img_run=0;
         if img_walk==5
             disp('walk')
             img_walk=0;
             img_run=0;
         else
         end
     else
     end

%       produce data to graph mean
     data_means=fopen('D:\psm_programming\data_analysis\means.txt','
r');
     plot_means=fscanf(data_means,'%f');
     graph_means=plot(plot_means,'m-p');
     title('means of every single image with image 1')
     xlabel('image number')
     ylabel('means of velocity')

 end
```

# LIST OF TABLES

# LIST OF FIGURES

MOTION DETECTION FOR PC BASED SECURITY SYSTEM BY USING OPTICAL FLOW

NUR NABILAH BT MOHAMAD HAMID

This thesis is submitted as partial fulfillment of the requirement
for the award of the
Bachelor of Electrical and Electronic Engineering

Faculty of Electrical and Electronic Engineering
UNIVERSITI MALAYSIA PAHANG

NOVEMBER, 2010